(54) **VIDEO SLOT GAMING MACHINE**
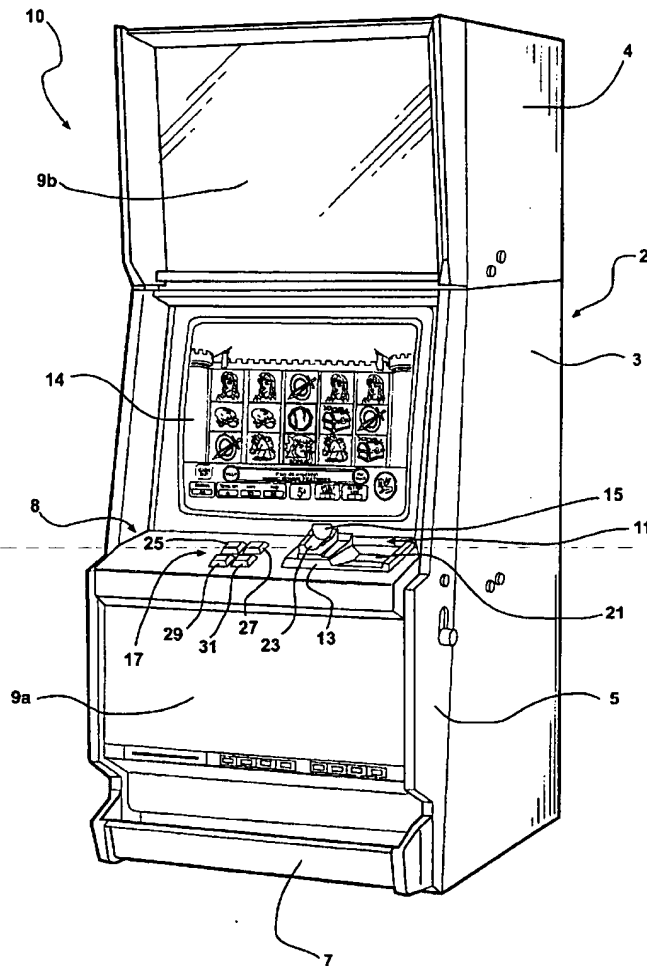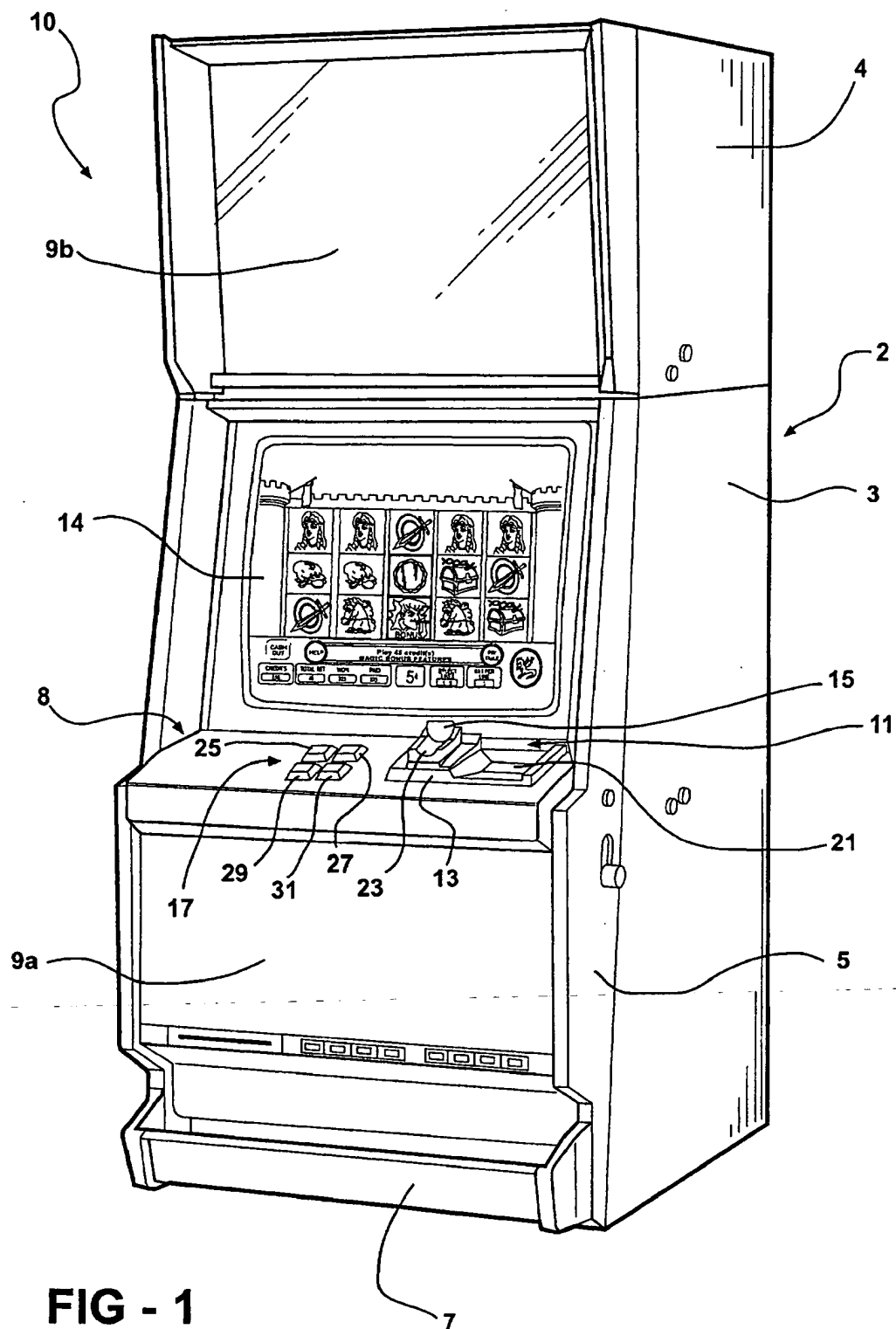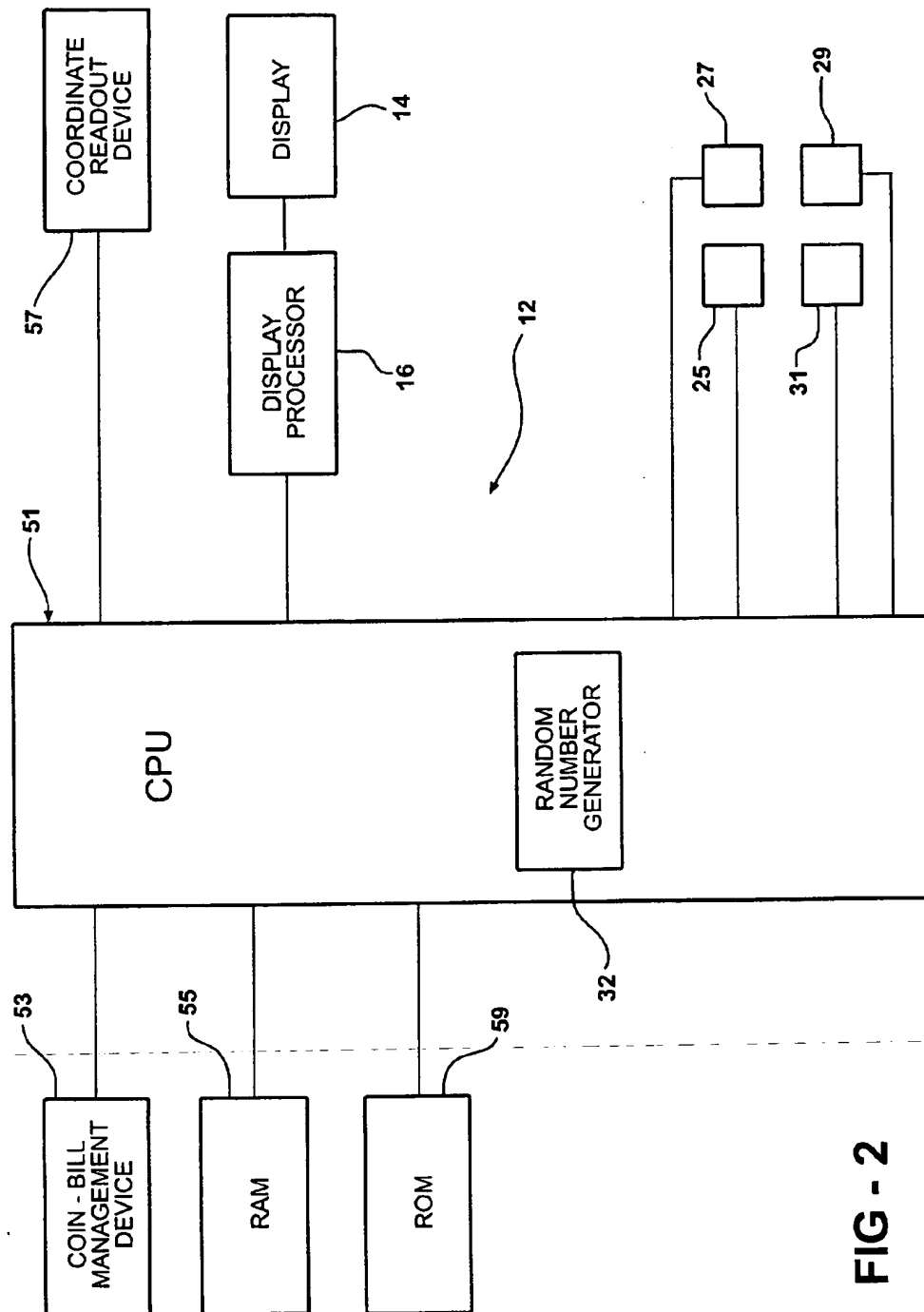
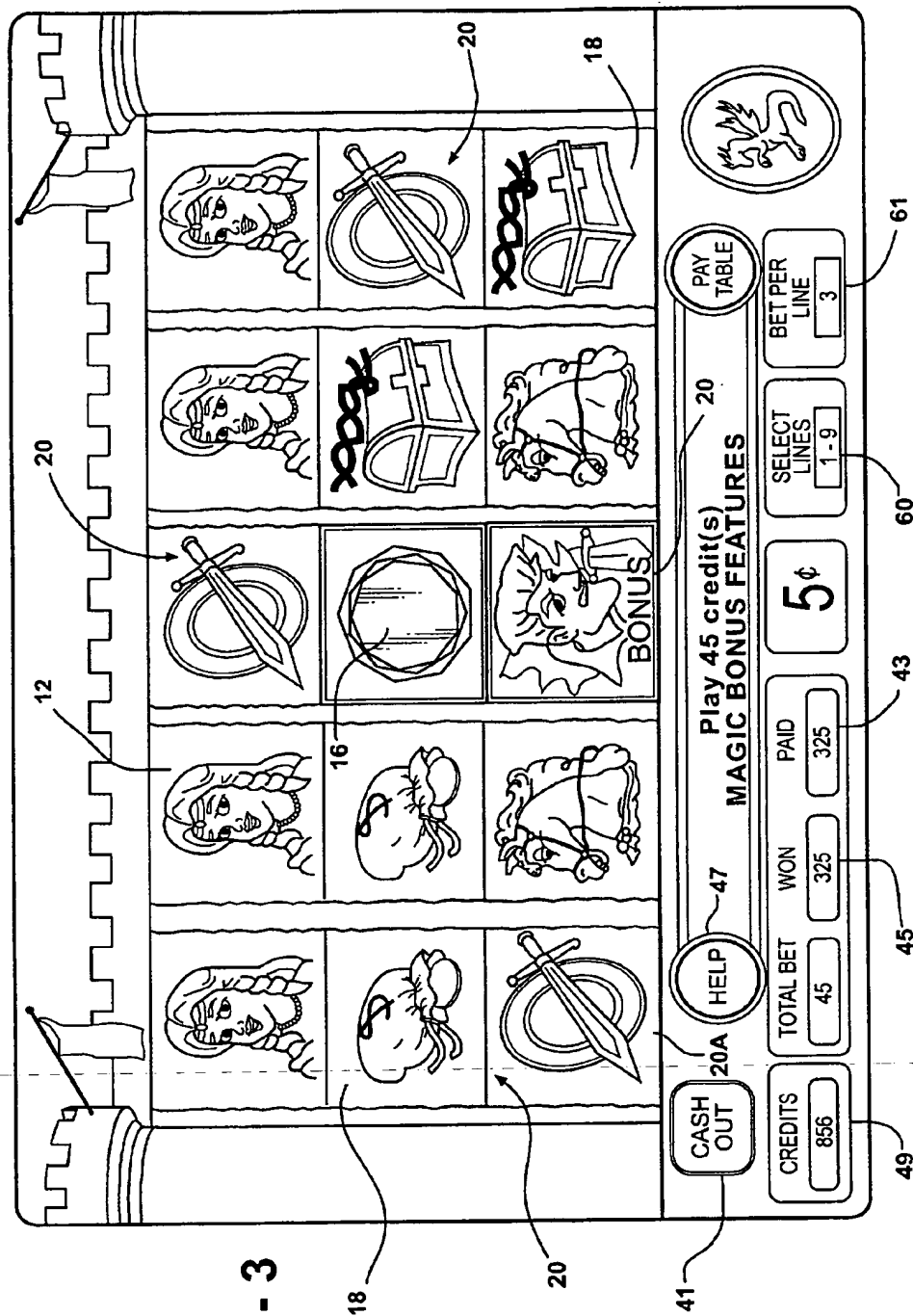(75) Inventor: Satoshi Suda, Atsugi-shi (JP)

Correspondence Address:
HOWARD & HOWARD ATTORNEYS, P.C.
THE PINEHURST OFFICE CENTER, SUITE #101
39400 WOODWARD AVENUE
BLOOMFIELD HILLS, MI 48304-5151 (US)

(73) Assignee: Konami Gaming, Inc., Las Vegas, NV

(57) **ABSTRACT**

A video slot machine includes a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns, a memory device for storing a pay-table, a game controller coupled to the display device and the memory device. The game controller being is adapted to randomly select the game elements to be displayed in the display device and to determine an outcome based on the displayed game elements, a pay-table, and predetermined paylines. The selected game elements are selected from a set of possible game elements. The set of possible game elements includes a bonus element. The game controller is adapted to identify the presence of the bonus element in one of the cells of a column and to modify all of the symbols within the column to wild if a wild character in any one cells of the column would modify the outcome. A bonus game is also provided which depicts a contest between the player and the game machine.

FIG - 1

FIG - 2

FIG - 3

FIG - 4

FIG - 5

—60

Randomly selecting a plurality of game elements, the selected game elements being select from a set of possible game elements, the set of possible game elements includes a bonus element.

—62

Displaying the plurality of game elements on a display device in a grid having a plurality of cells, defined by rows and columns.

—64

Determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines.

—66

Identifying the presence of the bonus element in one of the cells of a column.

—68

Modifying all of the symbols within the column to wild if a wild character in any one of the cells of the column would modify the outcome.

FIG - 6

72

Randomly selecting a plurality of game elements
from a set of possible game elements.

74

Displaying the plurality of game elements in a grid
having a plurality of cells, defined by rows and columns.

76

Determining an outcome based on the displayed game elements,
a pay-table, and predetermined paylines.

78

Activating a bonus game in response to detecting a triggering event,
the bonus game being depicted as a contest between the player
and the gaming machine.

80

Allowing the player to choose a player item from the three
possible player items.

82

Determining an outcome of the bonus game as a function of the
choosen player item

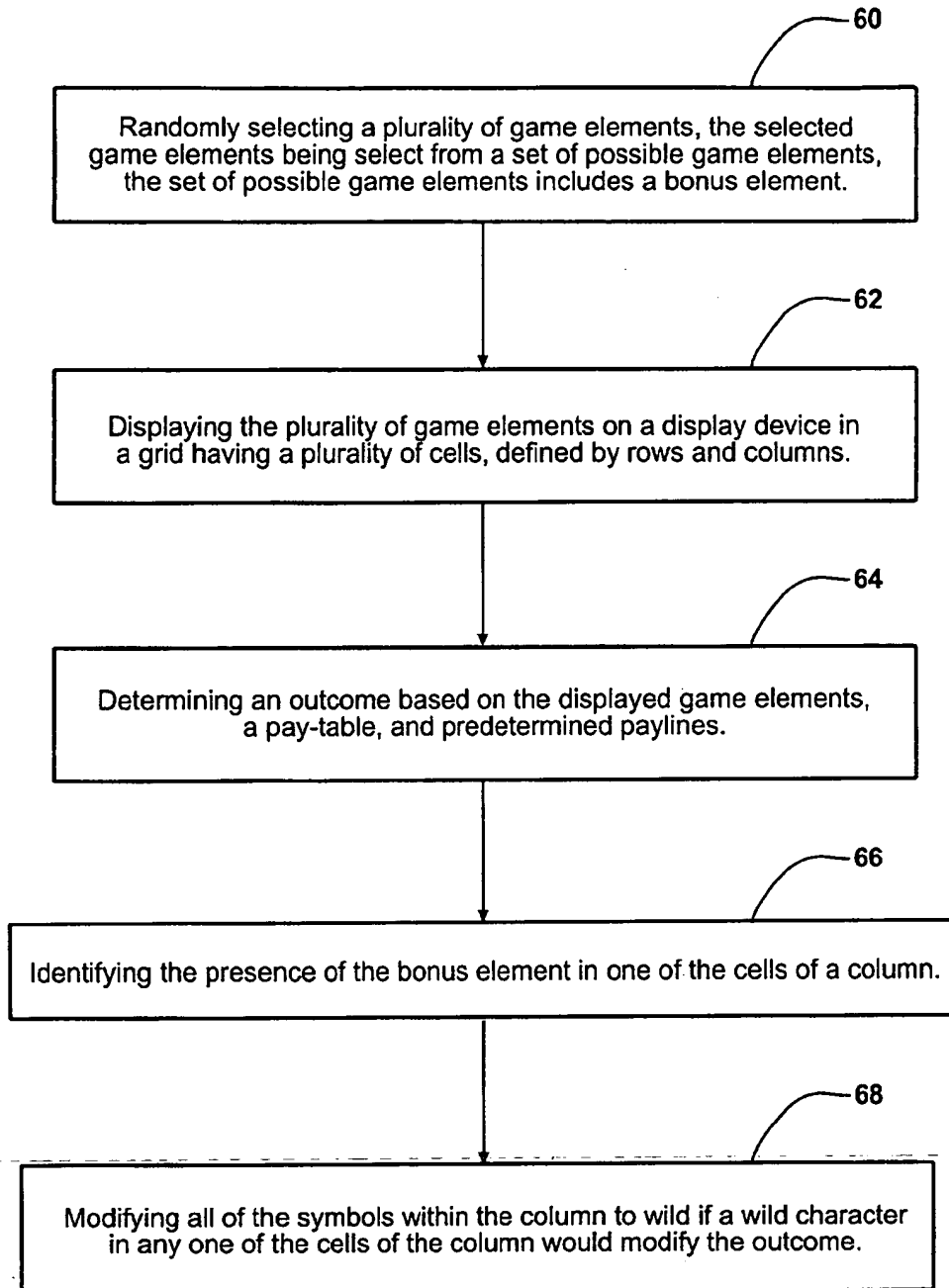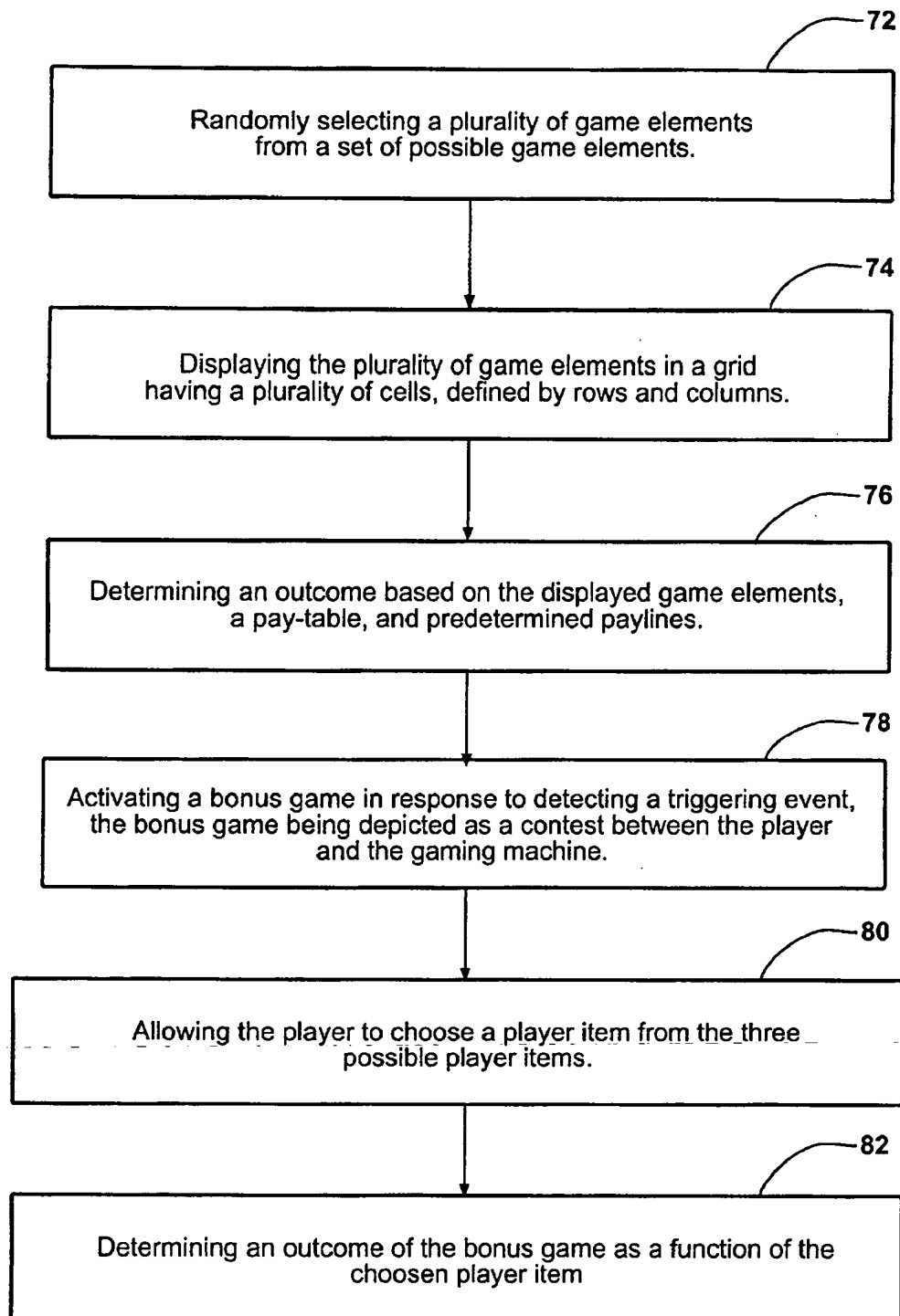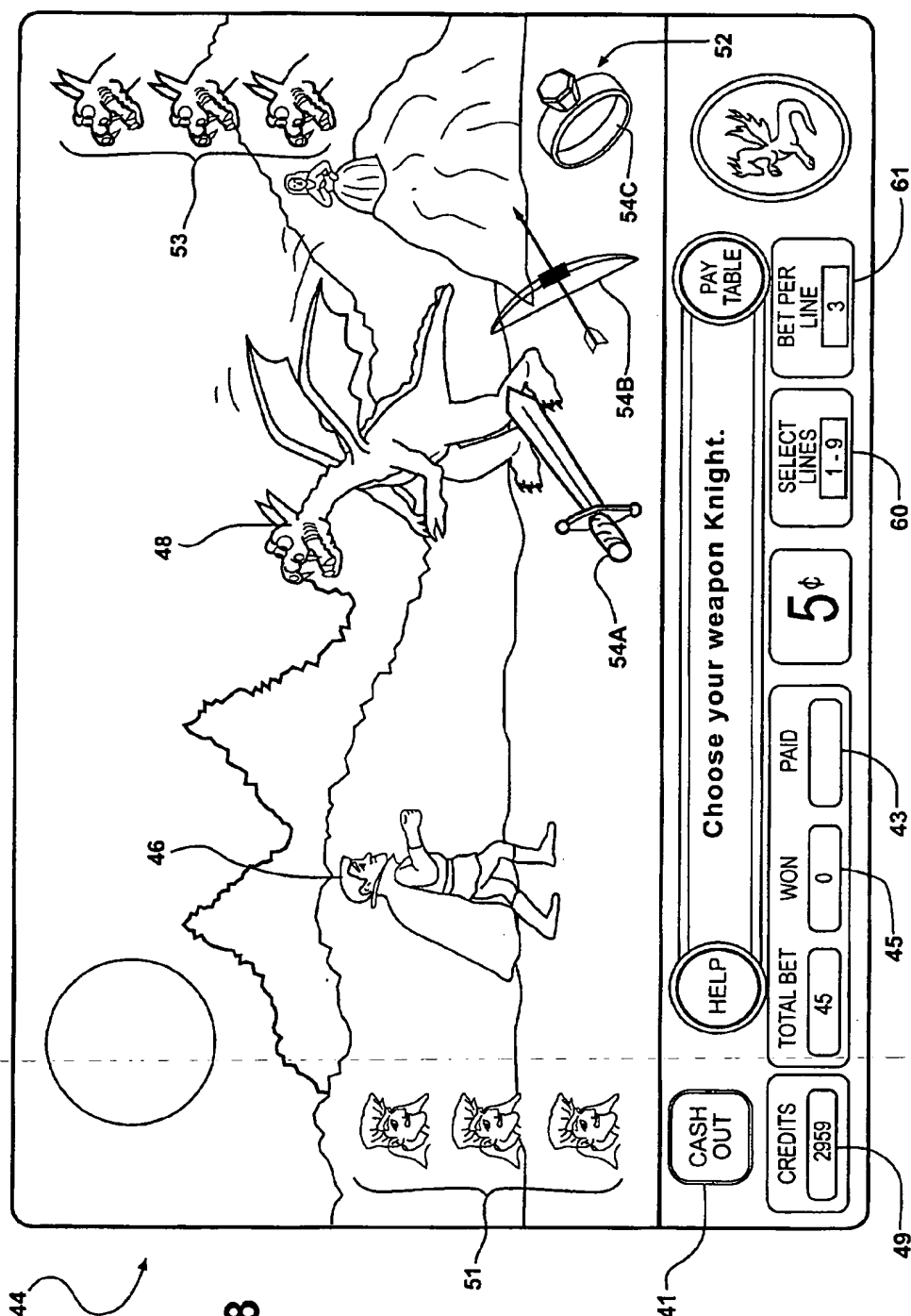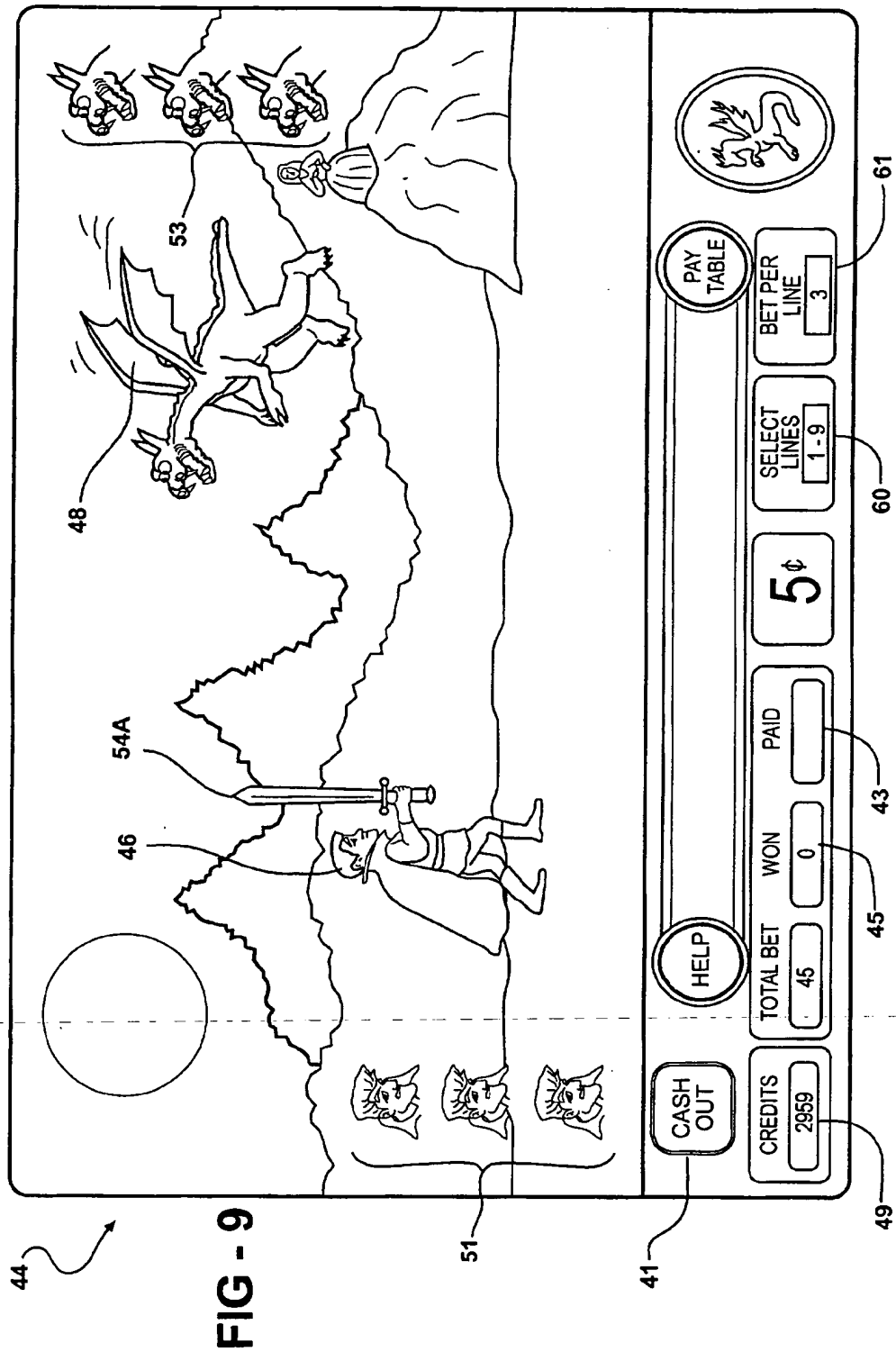# FIG - 7

FIG - 8

FIG - 9

VIDEO SLOT GAMING MACHINE

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates generally to video gaming machines and more particularly, to an apparatus and method for playing a video gaming machine having bonus games and a readable recording medium recording a control program for playing the video gaming machine having bonus games.

[0003] 2. Description of the Prior Art

[0004] Gaming machines, such as slot machines, are a cornerstone of the gaming industry. Generally, the popularity of such machines with players is dependent on the perceived likelihood of winning money at the particular game and the intrinsic entertainment value of the game relative to other available gaming options. Where the available gaming options include a number of competing games and the expectation of winning each game is believed to be generally the same, players are most likely to be attracted to the most entertaining and exciting games. Thus, gaming operators strive to employ the most entertaining and exciting games available because such games attract frequent play and, hence, increase profitability to the operator. Traditionally, a video gaming machine such as a slot machine includes a plurality of symbols including a wild symbol and a processor for randomly aligning the plurality of symbols on a display upon initiation of the game by a player. Generally, where the plurality of symbols are aligned so as to match a winning

[0005] combination of symbols stored in a pay-out table, the player receives a pay-out based on the wager placed by the player.

[0006] Such video gaming machine concepts are found, for instance, in U.S. Pat. No. 6,251,013 issued Jun. 26, 2001 in the name of Bennett The '013 patent discloses a video slot machine game in which a sprite randomly designates one or more of the symbols displayed on the display to be treated as special symbols.

[0007] Furthermore, one concept that has been successfully employed to enhance the entertainment value of the game is the addition of a bonus game that may be played in conjunction with the "primary" game. The bonus game may comprise any type of game, either similar to or completely different from the primary game. The bonus game is initiated upon the occurrence of a selected event or outcome of the primary game.

[0008] Because the excitement and entertainment value of the primary game provides increased player appeal relative to other gaming machines and the bonus game concept increases player appeal and excitement, thereby increasing the chance to win the potential pay-out amount, there is a continuing need to develop new features for primary and bonus games. New features are necessary to appeal to player interest and enhance excitement in order to entice longer play and satisfy demands of operators for interesting games and increased profitability. The present invention is directed to satisfying these needs.

SUMMARY OF THE INVENTION

[0009] In one aspect of the present invention, a video game machine is provided. The game machine includes a housing having a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns, a memory device for storing a pay-table, and a game controller coupled to the display device and the memory device. The game controller is adapted to randomly select the game elements to be displayed in the display device and to determine an outcome based on the displayed game elements, a wager, a pay-table, and predetermined paylines. The selected game elements are selected from a set of possible game elements. The set of possible game elements includes a bonus element. The game controller is adapted to identify the presence of the bonus element in one of the cells of a column and to modify all of the symbols within the column to wild if a wild character in any one cell of the column would modify the outcome.

[0010] In another aspect of the present invention, a video game machine is provided. The game machine includes a housing having a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns, a memory device for storing a pay-table, and a game controller coupled to the display device and the memory device. The game controller is adapted to randomly select the game elements to be displayed in the display device and to determine an outcome based on the displayed game elements, a wager, a pay-table, and predetermined paylines. The selected game elements are selected from a set of possible game elements. The game controller is adapted to activate a bonus game in response to detecting a triggering event. The bonus game is depicted as a contest between the gaming machine and the player, wherein the player chooses a player item from three possible player items in the bonus game. The outcome of the bonus game is determined as a function of the chosen player item. It should be noted that the number of possible items is not limited to three.

[0011] In yet another more aspect of the present invention, a method for playing a video gaming machine, is provided. The method includes the steps of randomly selecting a plurality of game elements and displaying the plurality of game elements on a display device in a grid having a plurality of cells defined by rows and columns. The selected game elements being selected from a set of possible game elements, the set of possible game elements includes a bonus element. The method further includes the steps of determining an outcome based on the displayed game elements, a wager, a pay-table, and predetermined paylines, identifying the presence of the bonus element in one of the cells of a column, and modifying all of the symbols within the column to wild if a wild character in any one cells of the column would modify the outcome.

[0012] An additional aspect of the present invention, a method for operating a video gaming machine for play by a player is provided. The method includes the steps of randomly selecting a plurality game elements from a set of possible game elements, displaying the plurality of game elements in a grid having a plurality of cells defined by rows and columns, and determining an outcome based on the displayed game elements, a wager, a pay-table, and predetermined paylines. The method further includes the steps of activating a bonus game in response to detecting a triggering event. The bonus game being depicted as a contest between the player and the game machine.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Other advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings wherein:

[0014] FIG. 1 is a perspective view of a gaming machine;

[0015] FIG. 2 is a schematic representation of the video gaming machine of the present invention;

[0016] FIG. 3 is a display of a plurality of elements including a bonus element in a first display during a normal random display having a winning combination appearing within the first display;

[0017] FIG. 4 is the display of FIG. 3 illustrating an animation, according to an embodiment of the present invention;

[0018] FIG. 5 is the display of FIG. 3 where the third column has been modified to wild elements;

[0019] FIG. 6 is a flow diagram illustrating operation of a video gaming machine, according to an embodiment of the present invention;

[0020] FIG. 7 is a flow diagram illustrating operation of a video gaming machine, according to another embodiment of the present invention;

[0021] FIG. 8 is a first graphical depiction of a bonus game, according to an embodiment of the present invention; and,

[0022] FIG. 9 is a second graphical depiction of the bonus game of FIG. 8.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0023] With reference to the drawings and in operation, the present invention provides a video slot gaming machine 10. With reference to FIG. 1, an exemplary video gaming machine 10 is illustrated into which the present invention can be incorporated to improve the enjoyment of a video game and to thereby increase the amount of time that the video game is played by patrons of a gaming establishment. FIG. 1 shows a general appearance of the video gaming machine 10 to which the present invention is applied. As shown in the FIG. 1, the machine 10 comprises a housing 2 standing upright. The housing 2 comprises a main body 3, a top box 4 mounted on a top portion of the main body 3 and a door 5 attached to a front side of the main body 3 so as to be swingable between an open position and a close position.

[0024] At a center portion of the front side of the main body 3, there is mounted a main display device or display 14 comprising a CRT, and below the display 14 is provided an operation panel 8. The operation panel 8 is attached to the door 5 so as to slope down in a forward direction of the machine 10. Below the operation panel 8 and on a front side of the top box 4, there are provided decoration panels 9a and 9b on which pictures, letters and the like representing a title of the machine 10 or the like are illustrated.

[0025] As shown in FIG. 1, the operation panel 8 is provided, from a right end toward a left end thereof, with an insertion portion 11, and an input portion 17. The insertion portion 11 is provided with a slot base 13 integrally formed with a coin insertion portion 15 and a bill insertion portion 21.

[0026] The input portion 17 is provided with four push button switches 25, 27, 29, 31 as first input devices, each of which is capable of being depressed. These push button switches 25, 27, 29, 31 are selected as switches to be operated with particular high frequency during the game, so that these switches are provided on the operation panel 8. For example, the push button switch 29 at a lower left position of the four switches is operated for starting the game. The number of the push buttons provided at the input portion 17 and functions assigned to the push buttons can be properly changed.

[0027] Referring now to FIG. 2, a block diagram illustrating a schematic configuration of a control system provided in the machine 10 is depicted. The machine 10 includes a game controller 12. The game controller 12 includes a central processing unit (CPU) 51, a coin-bill management device 53, a display processor 16, RAM 55 as a memory device and EPROM 59. The CPU 51 is mainly composed of a microprocessor unit and performs various calculations and motion control necessary for the progress of the game.

[0028] The coin-bill management device 53 detects the insertion of a coin and a bill from the coin insertion portion 15 and the bill insertion portion 21, and performs a necessary process for managing coins and bills. The display processor 16 interprets commands issued from the CPU 51 and displays desirable images on the display 14. The RAM 55 temporary stores programs and data necessary for the progress of the game, and the EPROM 59 stores, in advance, programs and data for controlling basic operation of the machine 10, such as the booting operation thereof.

[0029] The video gaming machine 10 of FIG. 1 further includes the display 14 that displays a video slot machine, or other game of chance, and a player using the video gaming machine 10 interacts with the game.

[0030] The CPU 51 is electrically connected with a coordinate readout device 57 as well as the above mentioned pushbutton switches 25, 27, 29, 31. The coordinate readout device 57 works as a second input device and comprises, for example, a so-called touch panel formed as a transparent panel on the display 14 and capable of issuing signals corresponding to the coordinates of a position touched on the display 14 by the player. The coordinate readout device 57 is closely put on the surface of the display 14 and integrated therewith. In the CPU 51, there are provided a payment processor 34 for counting value of money consumed in each game. A random number generator 32 is included for randomly generating the hand to be displayed in the game as described below.

[0031] With reference to FIGS. 1 and 3, the game controller 51 sends a signal to a display processor 16 for displaying a plurality of game elements 18 on the display 14. The display includes a cash-out touchpad 41 such that when the cash-out touchpad 41 is touched any accumulated credits are paid to the player in a coin bin 7. A winner paid meter 43 keeps track of credits paid out to a player. A credit meter 45 is displayed for informing the player of the number of winning credits won on a given spin. The touchpad could also be buttons affixed to the machine.

[0032] The display 14 further includes a help touchpad 47 for accessing information about the game. A credit meter 49 displays to the player a number of credits available to the player for game play or cash-out. A select lines touchpad 60 allows the player to toggle through and select the available sets of paylines. Preferably, the video slot gaming machine 10 is a multi-line game, i.e., the paylines include vertical paylines and/or diagonal pay-lines, and/or zig-zag paylines. A bet per line touchpad 61 allows the player to toggle to increase the bet per line a credit at a time (up to the maximum bet).

[0033] Returning to **FIG. 2**, the payment processor 34 is connected to the game controller 12 for awarding a regular payout in response to the game elements 18 displayed on the display 14 matching a winning combination along one of the paylines selected by the player.

[0034] With reference to **FIG. 3**, in one aspect of the present invention, the game controller 12, the display device or display 14 is adapted to display the plurality of game elements 18 in a grid 20 having a plurality of cells defined by rows and columns. The game EPROM provides a regular game and a bonus game. In the regular game, the game EPROM is adapted to randomly select the game elements 18 to be displayed in the display device 14. The selected game elements 18 are selected from a set of possible game elements, e.g., a treasure chest, bag of money, sword and shield, horse, flower, castle, etc. It should be noted that any type of symbols or game elements may be used. The game EPROM is adapted to determine an outcome of the regular game based on the displayed game elements 18, the pay-table, a wager, and predetermined paylines.

[0035] The game EPROM is adapted to include a bonus feature. The set of possible game elements includes a bonus element 16, which in the preferred embodiment, is a gem (see **FIG. 3**). After a regular game (see above), the game EPROM is adapted to identify the presence of the bonus element 16 in one of the cells of a column. As shown in **FIG. 3**, the game elements 18 in the display do not illustrate a winning combination of elements in any payline. Under the bonus feature, if the bonus element 16 appears in a cell, the EPROM is adapted to determine if changing all of the game elements 18 in the same column as the bonus element 16 to a wild element, i.e., the wild element is equal to any of the possible game elements to complete a payline, would change the outcome of the game.

[0036] For example, as shown in **FIG. 3**, the top row of game elements includes from left to right: a prince, a princess, a sword and shield, a princess and a princess. The sword and shield element is in the same column as the bonus element 16. If the sword and shield element were changed to a wild element, the top horizontal payline would include four princesses and the wild element. The wild element is interpreted as being equal to a princess. Thus, a winning combination would be found on the pay-line.

[0037] If changing the symbols in the column which includes the bonus symbol 16, modifies the outcome, then all the gaming elements 18 in that column are changed to wild elements and the outcome of the game is determined.

[0038] With reference to **FIGS. 4 and 5**, the EPROM is adapted to display an animation of a dragon 28 flying across the display 14 and breathing fire 30 on the bonus element 16. Afterwards, the entire column is displayed as fire 32 (see **FIG. 5**).

[0039] With reference to **FIG. 6**, in one embodiment of the present invention, a method of playing a video gaming machine, according to the present invention will now be discussed. At block 60, a plurality of game elements are randomly selected from a set of possible game elements. The set of possible game elements includes a bonus element. Next, at block 62, the plurality of game elements are displayed on a display device in a grid having a plurality of cells defined by rows and columns. Next at block 64, an outcome is determined based on the displayed game elements, a pay-table, a wager, and predetermined paylines. Next at block 66, the presence of the bonus element in one of the cells of a column is identified. Next at block 68, all of the symbols within the column are modified to wild if a wild character in any one cells of the column would modify the outcome.

[0040] In another aspect of the present invention, the EPROM is adapted to activate a bonus game in response to detecting a triggering event. In the preferred embodiment, the triggering event is the appearance on the display 14 of a bonus game element 20, such as a gem, in **FIG. 3**.

[0041] In one embodiment, in **FIG. 8**, the bonus game is depicted as a contest between the gaming machine 10 and the player. The player is given the option to choose a player item 54a-c from three possible player items 54. The number of possible player items is not limited to three.

[0042] The EPROM is adapted to randomly select a game item from three computer items in the bonus game. The outcome of a bonus game combat, i.e., whether the player wins or loses the bonus game is determined as a function of the chosen player item, the chosen game item, and a set of predetermined rules. If the player wins the bonus game combat round, the player wins a bonus or prize.

[0043] In one embodiment, the player and the gaming machine 10 are given three lives, Prince lives 51 and Dragon lives 53. The bonus game continues until either the player or the gaming machine 10 have zero lives. Each round of the bonus game, the player or the gaming machine 10 or both lose a life based on the predetermined rules. The predetermined rules are a given combination of the player item and the game item, the result is determined by computer preset rules. The set of predetermined rules includes three outcomes: player wins, player loses, and tie. The gaming machine 10 loses a life if the player wins, the player loses a life if the player loses, and the layer and the gaming machine 10 both lose a life if there is a tie.

[0044] Preferably, the player wins the bonus game, and is awarded the bonus credits, unless the player reaches zero lives before the gaming machine, i.e., if the gaming machine 10 reaches zero before or at the same time as the player.

[0045] If the player wins the bonus game, the game is adapted to determine a bonus payout and to distribute the bonus payout to the player.

[0046] Additionally, the game is adapted to display an animation of the contest between the gaming machine 10 and the player on the display device 14.

[0047] With reference to **FIG. 7**, in another embodiment of the present invention, a method for operating a video gaming machine 10 for play by a player will now be discussed. In a first process block 72, a plurality of game

elements are randomly selected from a set of possible game elements. In a second process block **74**, the plurality of game elements are displayed in a grid having a plurality of cells defined by rows and columns. In a third process block **76**, an outcome is determined based on the displayed game elements, a pay-table, and predetermined paylines. In a fourth process block **78**, a bonus game is activated in response to detecting a triggering event. The bonus game is depicted as a contest between the player and the gaming machine **10**. In a fifth process block **80**, the player is allowed to choose a player Prince's weapon item from three possible player weapons. In a sixth process block **82**, an outcome of the bonus game is determined as a function of the chosen player item, vis-a-vis the Dragon's combat stance.

[0048] With reference to **FIGS. 8 and 9**, in one embodiment, the player is represented by a hero or prince **46** and the gaming machine **10** is represented by an enemy or dragon **48**. The contest is a fight between the prince **46** and the dragon **48**. The game controller **12** is adapted to display an animation representing the fight between the prince **46** and the dragon **48**.

[0049] As shown in **FIG. 8**, the player is given a choice of three weapons: a sword **54A**, a bow and arrow **54B**, and a magic ring **54C**, which may be selected by touching the corresponding video representation on the display **14**. At the start of the bonus game, the prince **46** and the dragon **48** are each given three lives, as indicated by the prince icons **51** and the dragon icons **53**.

[0050] The computer items from which the game controller **12** selects includes: a dragon air attack, a dragon ground attack, and a dragon magic tornado attack. The air attack is shown in **FIG. 9**.

[0051] For example, if the player chooses the sword **54A** and the game controller **12** chooses the dragon attack, there is a tie. If the game controller **12** chose the dragon flight attack, the player loses. If the game controller **12** chose the dragon tornado attack, the player wins.

[0052] Preferably, if during the combat or bonus round the prince wins or ties a round or melee, the player gets a first bonus. Further at end of the bonus, the player gets a low bonus if the prince dies (loses all of his lives before the dragon). If the prince and dragon tie, or the prince wins, the player gets a higher bonus.

[0053] Additionally, a different animation is shown depending on the result of the bonus round. For example, if the prince wins, the prince saves the princess in the animation. If the prince loses, the dragon takes the princess away.

[0054] Other aspect and features of the present invention can be obtained from a study of the drawings, the disclosure, and the appended claims.

What is claimed is:

1. A video gaming machine, comprising:

a housing having a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns;

a memory device for storing a pay-table; and,

a game controller coupled to the display device and the memory device, the game controller being adapted to randomly select the game elements to be displayed in the display device, the selected game elements being selected from a set of possible game elements, and to determine an outcome based on the displayed game elements, a pay-table, and predetermined paylines,

wherein the set of possible game elements includes a bonus element, and wherein the game controller is adapted to identify the presence of the bonus element in one of the cells of at least one column and to modify all of the symbols within the column to wild if a wild character in any one cells of the column would modify the outcome.

2. A video gaming machine, as set forth in claim 1, wherein the game controller is adapted to display an animation of an enemy breathing fire on the bonus element, changing the symbols in the column to fire to symbolize the changing of the symbols to wild characters.

3. A video gamin machine, as set forth in claim 1, wherein the bonus element is a jewel.

4. A method for playing a video gaming machine, including the steps of:

randomly selecting a plurality of game elements, the selected game elements being selected from a set of possible game elements, the set of possible game elements includes a bonus element;

displaying the plurality of game elements on a display device in a grid having a plurality of cells defined by rows and columns;

determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines;

identifying the presence of the bonus element in one of the cells of a column; and,

modifying all of the symbols within the column to wild if a wild character in any one cells of the column would modify the outcome.

5. A video gaming machine for play by a player, comprising:

a housing having a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns;

a memory device for storing a pay-table; and,

a game controller coupled to the display device and the memory device, the game controller being adapted to randomly select the game elements to be displayed in the display device, the selected game elements being selected from a set of possible game elements, and to determine an outcome based on the displayed game elements, a pay-table, and predetermined paylines, the game controller being adapted to activate a bonus game in response to detecting a triggering event, the bonus game being depicted as a fight based between a hero and an enemy, wherein the player chooses a weapon from three weapons and the outcome of the bonus game is determined as a function of the chosen weapon.

6. A video gaming machine, as set forth in claim 5, wherein the three weapons are a sword, a bow and arrow, and a magic ring.

7. A video gaming machine, as set forth in claim 6, wherein the game controller is adapted to randomly select an enemy attack from three enemy attacks, wherein the out-

come of the bonus game is determined as a function of the chosen weapon, the chosen enemy attack, and a set of predetermined rules.

8. A video gaming machine, as set forth in claim 7, wherein the three enemy attacks are attack from the air, attack from the ground, and a magic tornado attack.

9. A video gaming machine, as set forth in claim 8, wherein the set of predetermined rules includes three outcomes: hero wins, hero loses, and tie.

10. A video gaming machine, as set forth in claim 9, wherein the enemy and the hero each have three lives.

11. A video gaming machine, as set forth in claim 10, wherein the enemy loses a life if the hero wins, the hero loses a life if the hero loses, and the hero and the enemy both lose a life if there is a tie.

12. A video gaming machine, as set forth in claim 11, wherein the player wins the bonus game unless the player reaches zero lives before the enemy.

13. A video gaming machine, as set forth in claim 12, wherein the game controller is adapted to determine a bonus payout if the player wins and to distribute the bonus payout to the player.

14. A video gaming machine, as set forth 6, wherein the game controller is adapted to display an animation of the fight between the hero and the enemy on the display device.

15. A method for operating a video gaming machine for play by a player, including the steps of:

randomly selecting a plurality game elements from a set of possible game elements;

displaying the plurality of game elements in a grid having a plurality of cells defined by rows and columns;

determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines;

activating a bonus game in response to detecting a triggering event, the bonus game being depicted as a fight based between a hero and an enemy;

allowing the player to choose a weapon from three weapons; and,

determining an outcome of the bonus game as a function of the chosen weapon.

16. A method, as set forth in claim 15, wherein the three weapons are a sword, a bow and arrow, and a magic ring.

17. A method, as set forth in claim 16, including the step of randomly selecting an enemy attack from three enemy attacks, wherein the outcome of the bonus game is determined as a function of the chosen weapon, the chosen enemy attack, and a set of predetermined rules.

18. A method, as set forth in claim 17, wherein the three enemy attacks are attack from the air, attack from the ground, and a magic tornado attack.

19. A method, as set forth in claim 18, wherein the set of predetermined rules includes three outcomes: hero wins, hero loses, and tie.

20. A method, as set forth in claim 19, wherein the enemy and the hero each have three lives.

21. A method, as set forth in claim 20, wherein the enemy loses a life if the hero wins, the hero loses a life if the hero loses, and the hero and the enemy both lose a life if there is a tie.

22. A method, as set forth in claim 21, wherein the player wins the bonus game unless the player reaches zero lives before the enemy.

23. A method, as set forth in claim 22, including the steps of:

determining a bonus payout if the player wins; and,

distributing the bonus payout to the player.

24. A method, as set forth in claim 15, including the step of providing an animation of the fight between the hero and the enemy.

25. A video gaming machine for play by a player, comprising:

a housing having a display device for displaying a plurality of game elements in a grid having a plurality of cells defined by rows and columns;

a memory device for storing a pay-table; and,

a game controller coupled to the display device and the memory device, the game controller being adapted to randomly select the game elements to be displayed in the display device, the selected game elements being selected from a set of possible game elements, and to determine an outcome based on the displayed game elements, a pay-table, and predetermined paylines, the game controller being adapted to activate a bonus game in response to detecting a triggering event, wherein the bonus game is depicted as a contest between the gaming machine and the player, wherein the player chooses a player item from three possible player items in the bonus game and the outcome of the bonus game is determined as a function of the chosen player item.

26. A video gaming machine, as set forth in claim 25, wherein the game controller is adapted to randomly select a game item from three computer items in the bonus game, wherein the outcome of the bonus game is deter mined as a function of the chosen player item, the chosen game item, and a set of predetermined rules.

27. A video gaming machine, as set forth in claim 26, wherein the set of predetermined rules includes three outcomes: player wins, player loses, and tie.

28. A video gaming machine, as set forth in claim 27, wherein the gaming machine and the player each have three lives.

29. A video gaming machine, as set forth in claim 28, wherein the gaming machine loses a life if the player wins, the player loses a life if the player loses, and the layer and the gaming machine both lose a life if there is a tie.

30. A video gaming machine, as set forth in claim 29, wherein the player wins the bonus game unless the player reaches zero lives before the gaming machine.

31. A video gaming machine, as set forth in claim 30, wherein the game controller is adapted to determine a bonus payout if the player wins and to distribute the bonus payout to the player.

32. A video gaming machine, as set forth 25, wherein the game controller is adapted to display an animation of the contest between the gaming machine and the player on the display device.

33. A method for operating a video gaming machine for play by a player, including the steps of:

randomly selecting a plurality of game elements from a set of possible game elements;

displaying the plurality of game elements in a grid having a plurality of cells defined by rows and columns;

determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines;

activating a bonus game in response to detecting a triggering event, the bonus game being depicted as a contest between the player and the gaming machine;

allowing the player to choose a player item from three possible player items; and,

determining an outcome of the bonus game as a function of the chosen player item.

34. A method, as set forth in claim 33, including the step of randomly selecting a computer item from three possible computer items, wherein the outcome of the bonus game is determined as a function of the chosen player item, the chose computer item, and a set of predetermined rules.

35. A method, as set forth in claim 34, wherein the set of predetermined rules includes three outcomes: player wins, player loses, and tie.

36. A method, as set forth in claim 19, wherein the player and the gaming machine each have three lives.

37. A method, as set forth in claim 20, wherein the gaming machine loses a life if the player wins, the player loses a life if the player loses, and the player and the gaming machine both lose a life if there is a tie.

38. A method, as set forth in claim 21, wherein the player wins the bonus game unless the player reaches zero lives before the gaming machine.

39. A method, as set forth in claim 22, including the steps of:

determining a bonus payout if the player wins; and,

distributing the bonus payout to the player.

40. A method, as set forth in claim 15, including the step of providing an animation of the contest between the player and the gaming machine.

41. A computer program product for playing a video gaming machine, the computer program product comprising a computer usable storage medium having computer readable program code means embodied in the medium, the computer readable program code means comprising:

computer readable program code means for randomly selecting a plurality of game elements, the selected game elements being selected from a set of possible game elements, the set of possible game elements includes a bonus element;

computer readable program code means for displaying the plurality of game elements on a display device in a grid having a plurality of cells defined by rows and columns;

computer readable program code means for determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines;

computer readable program code means for identifying the presence of the bonus element in one of the cells of a column; and,

computer readable program code means for modifying all of the symbols within the column to wild if a wild character in any one cells of the column would modify the outcome.

42. A computer program product for playing a video gaming machine, the computer program product comprising a computer usable storage medium having computer readable program code means embodied in the medium, the computer readable program code means comprising:

computer readable program code means for randomly selecting a plurality of game elements from a set of possible game elements;

computer readable program code means for displaying the plurality of game elements in a grid having a plurality of cells defined by rows and columns;

computer readable program code means for determining an outcome based on the displayed game elements, a pay-table, and predetermined paylines; and,

computer readable program code means for activating a bonus game in response to detecting a triggering event, the bonus game being depicted as a contest between the player and the gaming machine.

* * * * *

(12) **United States Patent** (10) **Patent No.:** **US 6,364,765 B1**

Walker et al. (45) **Date of Patent:** **Apr. 2, 2002**

(54) **ELECTRONIC AMUSEMENT DEVICE OFFERING SECONDARY GAME OF CHANCE AND METHOD FOR OPERATING SAME**

(75) Inventors: **Jay S. Walker**, Ridgefield; **James A. Jorasch**, Stamford; **Magdalena Mik**, Greenwich; **Stephen C. Tulley**, Stamford; **Robert R. Lech**, Norwalk, all of CT (US)

(73) Assignee: **Walker Digital, LLC**, Stamford, CT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/108,646**

(22) Filed: **Jul. 1, 1998**

(51) Int. Cl.^{7} A63F 13/00; A63F 9/24; G06F 17/00; G06F 19/00

(52) U.S. Cl. 463/16; 463/17; 463/18; 463/19; 463/20; 463/12; 463/13; 273/138.1; 273/237; 273/269; 273/297

(58) Field of Search .... 463/16, 17, 18, 463/19, 20, 12, 13, 30, 37, 40, 41, 42, 43; 273/138.1, 142 B, 297, 293, 142 R, 143 R, 237, 269

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,856,787	A	*	8/1989	Itkis .... 273/237 X
5,083,271	A		1/1992	Thacher et al.
5,179,517	A		1/1993	Sarbin et al.
5,242,163	A	*	9/1993	Fulton .... 273/85 CP X
5,342,047	A	*	8/1994	Haidel et al. .... 273/85 CP X
5,393,057	A	*	2/1995	Marnell, II .... 273/85 CP X
5,409,225	A		4/1995	Kelly et al.
5,564,700	A		10/1996	Celona
5,580,309	A		12/1996	Piechowiak et al.
5,611,730	A		3/1997	Weiss
5,639,088	A	*	6/1997	Schneider et al. ... 273/138.2 X
5,702,304	A		12/1997	Acres et al.
5,848,932	A	*	12/1998	Adams .... 463/20 X
5,979,702	A	*	11/1999	Hennessey et al. ..... 221/206 X
5,980,384	A	*	11/1999	Barrie .... 463/16 X
6,033,307	A	*	3/2000	Vancura .... 463/20 X

* cited by examiner
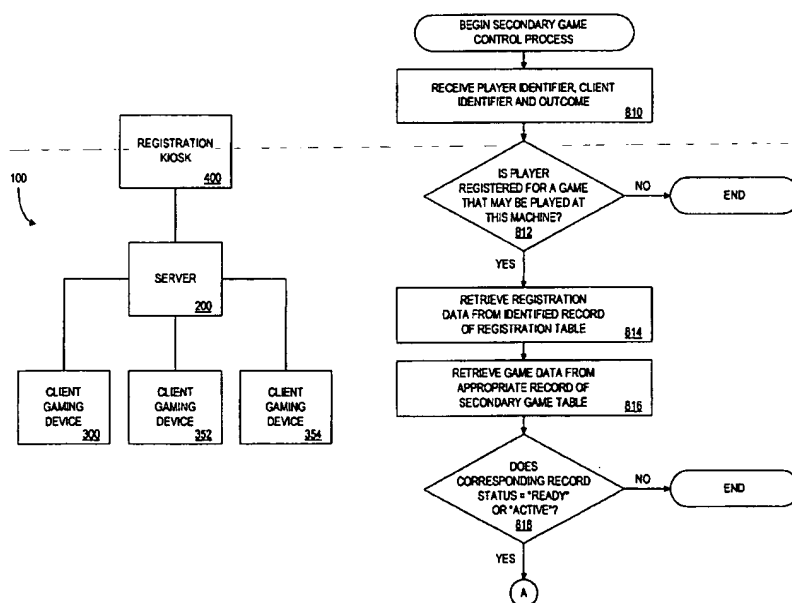
Primary Examiner—Joe H. Cheng
Assistant Examiner—Binh-An D. Nguyen
(74) *Attorney, Agent, or Firm*—Dean P. Alderucci

(57) **ABSTRACT**

An electronic amusement device and a method for operating the device are disclosed. In accordance with the present invention, a slot machine server controls a secondary game of chance played at a client slot machine. The server receives a player identifier corresponding to a player and a client identifier corresponding to a client slot machine. The server retrieves registration data relating to a secondary game of chance corresponding to the player identifier and the client identifier. The server further receives an outcome from the client slot machine and analyzes the outcome based on game requirements associated with the secondary game of chance, thereby determining whether the outcome satisfies at least one of the game requirements. Once all of the game requirements have been satisfied, the server updates a session status indicating the completion of the game requirements.

43 Claims, 19 Drawing Sheets

100

REGISTRATION
KIOSK

<u>400</u>

SERVER

<u>200</u>

CLIENT
GAMING
DEVICE

<u>300</u>

CLIENT
GAMING
DEVICE

<u>352</u>

CLIENT
GAMING
DEVICE

<u>354</u>

FIG. 1

200 →

```
REGISTRATION
KIOSK
400
```

```
CLOCK
212
```

```
PROCESSOR
210
```

```
COMMUNICATION
PORT
220
```

```
CLIENT
GAMING
DEVICE
300
```

```
CLIENT
GAMING
DEVICE
352
```

```
CLIENT
GAMING
DEVICE
354
```

214

216
```
PROGRAM
```

218
```
PLAYER TABLE
```

500
```
SECONDARY GAME TABLE
```

600
```
REGISTRATION TABLE
```

700
```
GAME OUTCOME TABLE
```

FIG. 2

FIG. 3

400

CURRENCY ACCEPTOR 414	DISPLAY 416	CARD READER 418

PRINTER 422	PROCESSOR 410	COMMUNICATION PORT 412	SERVER 200

HOPPER CONTROLLER 424		INPUT DEVICE 420

HOPPER 426

428

PROGRAM 430

FIG. 4

500

GAME IDENTIFIER 510	NUMBER OF PLAYERS 512	GAME TYPE 514	GAME TIME LIMIT 516	GAME PLAY LIMIT 518	
ABC	1	SOLO	N/A	225 PLAYS	
BCD	1	SOLO	45 MINS.	N/A	
CDE	1	SOLO	N/A	150 PLAYS	
DEF	2	COMBINATION	60 MINS.	N/A	
EFG	6	COMPETITION	N/A	N/A	
FGH	3	COMBINATION	30 MINS.	N/A	

550

560

570

FIG. 5A

500 (CONT.) ⟶

GAME BONUS AMOUNT 520	CLIENT IDENTIFIERS 522	GAME REQUIREMENTS 524
$25	YELLOW	COLLECT 100 LEMONS
$15	YELLOW BLUE RED	OBTAIN CHERRY-CHERRY-CHERRY COMBINATION ON 1 YELLOW MACHINE, 1 BLUE MACHINE, AND 1 RED MACHINE
$30	BLUE	OBTAIN 6 JACKPOTS WITH PAYOUTS > 20 TOKENS EACH WITHIN 150 PLAYS
$50	BLUE RED DUECES WILD ANY VIDEO POKER	1.) CHERRY-CHERRY-CHERRY ON ANY BLUE MACHINE 2.) FLUSH HAND ON ANY VIDEO POKER MACHINE 3.) ANY WINNING OUTCOME ON ANY RED MACHINE 4.) 10 ACE OF SPADES ON ANY DEUCES WILD VIDEO POKER MACHINE
$20	ANY SLOT ANY VIDEO POKER BLUE	1.) ANY PAIR JACKS OR BETTER ON ANY VIDEO POKER MACHINE 2.) CHERRY-CHERRY-CHERRY ON ANY 3 REEL SLOT MACHINE 3.) 3 OF A KIND ON ANY VIDEO POKER MACHINE 4.) ORANGE-ORANGE-ORANGE ON ANY BLUE MACHINE WHILE WAGERING MAXIMUM AMOUNT
$20	YELLOW BLUE RED	1.) 10 CHERRIES ON AT LEAST ONE BLUE MACHINE 2.) 10 LEMONS ON AT LEAST ONE RED MACHINE 3.) 10 BARS ON AT LEAST ONE RED MACHINE

FIG. 5B

— 600

SESSION IDENTIFIER: 0326981418				610
GAME IDENTIFIER: ABC				612
PLAYER IDENTIFIER 614	TIME REMAINING 616	PLAYS REMAINING 618	SESSION STATUS 620	BONUS DUE 622
24681012	N/A	105	ACTIVE	$0

650

700 —

SESSION IDENTIFIER: 0326981418			710
PLAYER IDENTIFIER 711	CLIENT IDENTIFIER 712	OUTCOME 714	TIMESTAMP 716
24681012	SM 1002003	LEMON	3/26/98 03:15:30 PM
24681012	SM 1001098	LEMON	3/26/98 03:19:23 PM

750

752

FIG. 6A

— 600

SESSION IDENTIFIER: 0327981148				610
GAME IDENTIFIER: DEF				612
PLAYER IDENTIFIER 614	TIME REMAINING 616	PLAYS REMAINING 618	SESSION STATUS 620	BONUS DUE 622
36917154	60:00 MIN.	N/A	READY	$0
48129003	60:00 MIN.	N/A	READY	$0

652

654

700 —

SESSION IDENTIFIER: 0327981148			710
PLAYER IDENTIFIER 711	CLIENT IDENTIFIER 712	OUTCOME 714	TIMESTAMP 716

FIG. 6B

600

SESSION IDENTIFIER: 0328981012				610
GAME IDENTIFIER: EFG				612
PLAYER IDENTIFIER 614	TIME REMAINING 616	PLAYS REMAINING 618	SESSION STATUS 620	BONUS DUE 622
35791130	N/A	N/A	SUCCESS	$20
24483696	N/A	N/A	FAIL	$0

656
658

700

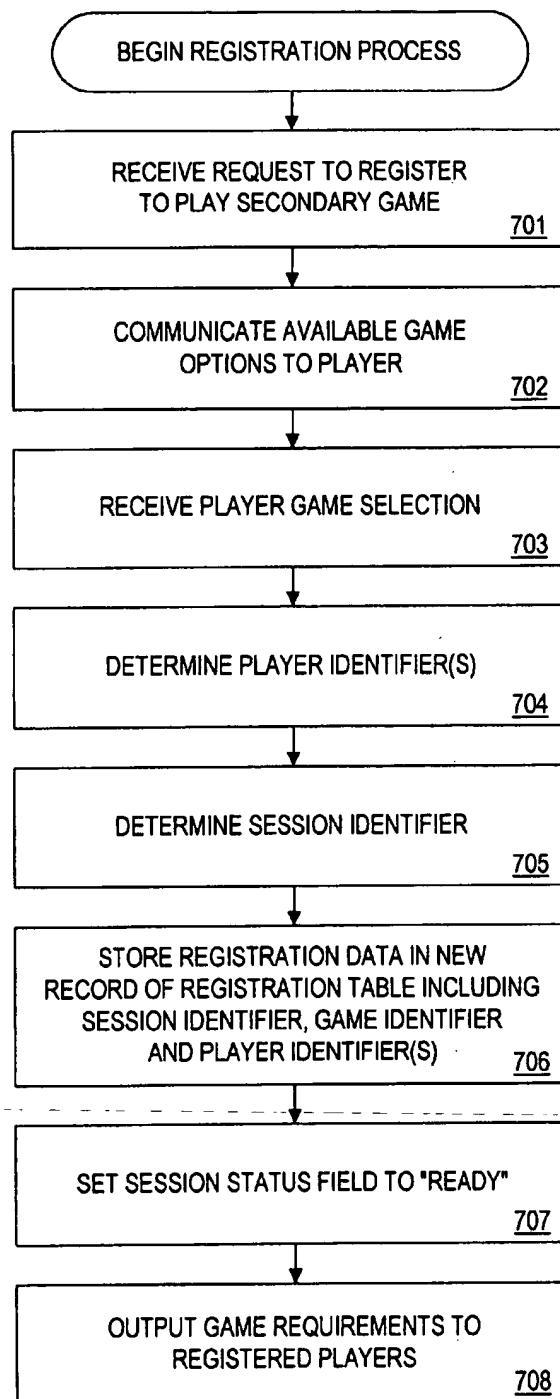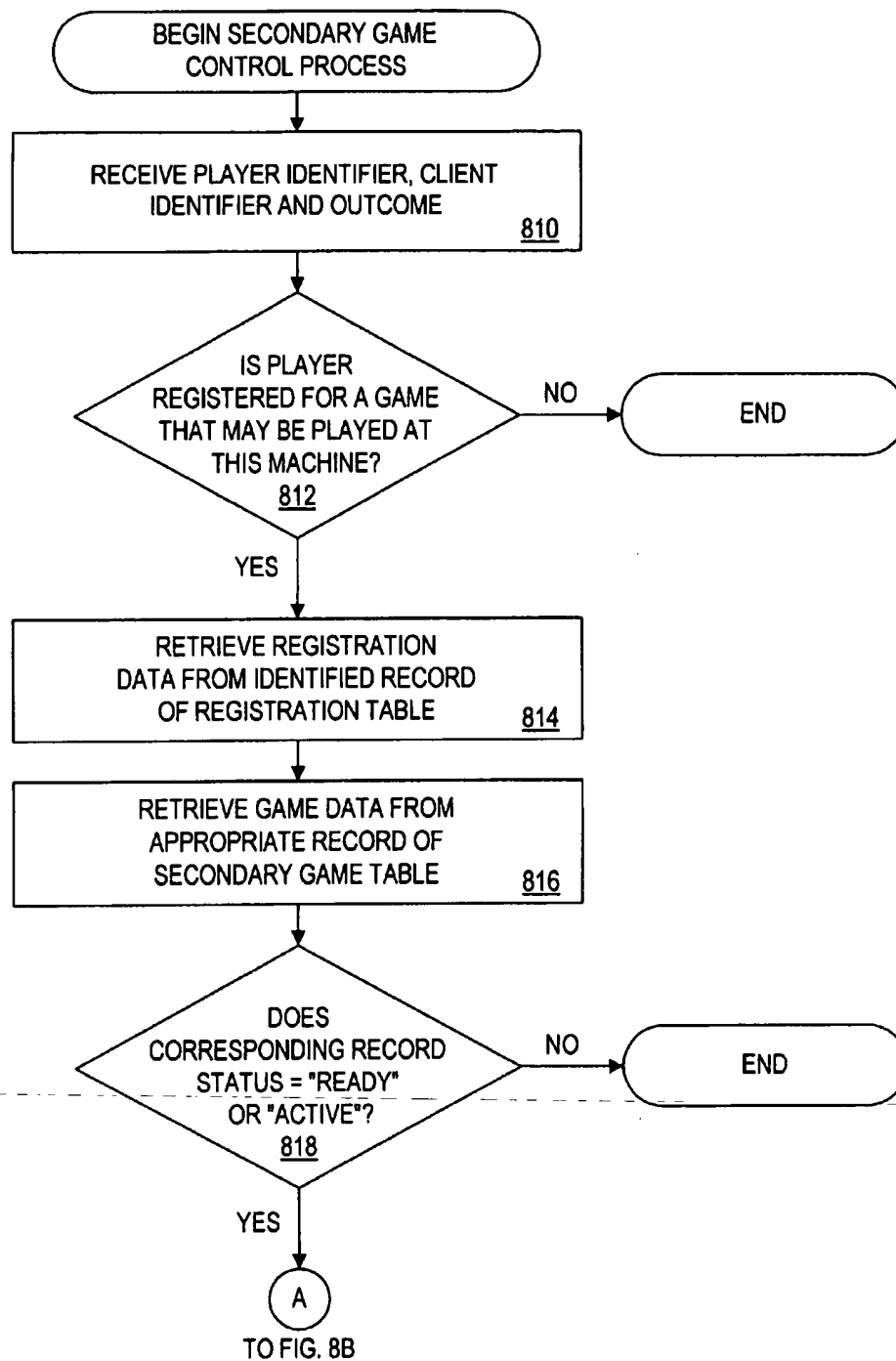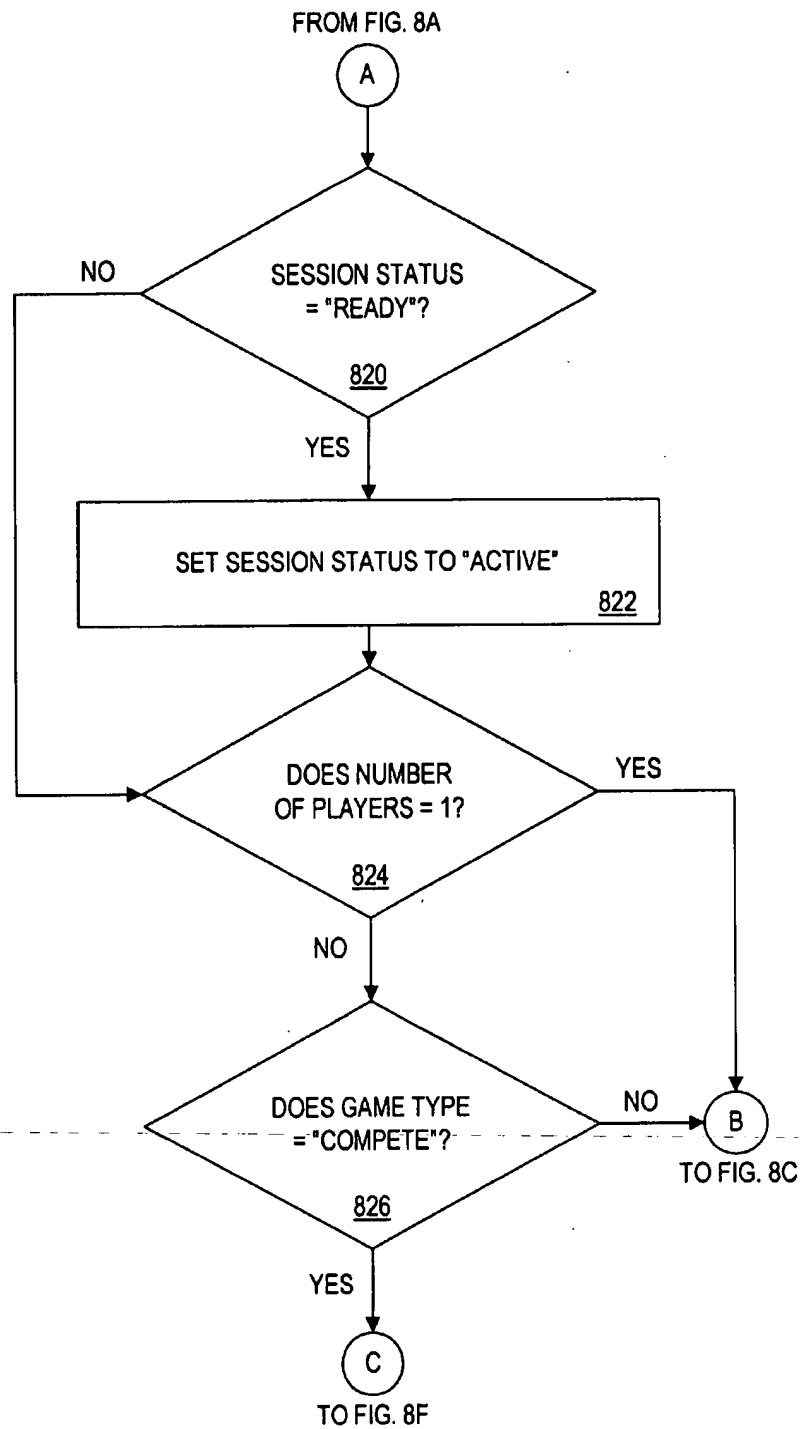| SESSION IDENTIFIER: 0328981012			710
PLAYER IDENTIFIER 711	CLIENT IDENTIFIER 712	OUTCOME 714	TIMESTAMP 716
35791130	VP 8423456	PAIR Q'S	3/28/98 10:15:21 AM
24483696	SM 15874953	CH - CH - CH	3/28/98 11:41:39 AM
24483696	VP 8423456	THREE 4'S	3/28/98 12:38:46 PM
35791130	SM B657423	OR - OR - OR	3/28/98 1:51:22 PM
24483696	VP 8659187	PAIR K'S	3/28/98 2:47:18 PM
35791130	SM B654213	CH - CH - CH	3/28/98 3:21:45 PM
35791130	VP 8659187	THREE ACES	3/28/98 4:10:17 PM

754
756
758
760

FIG. 6C

```
        ┌─────────────────────────────────┐
        │   BEGIN REGISTRATION PROCESS    │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  RECEIVE REQUEST TO REGISTER    │
        │  TO PLAY SECONDARY GAME         │
        │                           701   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  COMMUNICATE AVAILABLE GAME     │
        │  OPTIONS TO PLAYER              │
        │                           702   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  RECEIVE PLAYER GAME SELECTION  │
        │                           703   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  DETERMINE PLAYER IDENTIFIER(S) │
        │                           704   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  DETERMINE SESSION IDENTIFIER   │
        │                           705   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  STORE REGISTRATION DATA IN NEW │
        │  RECORD OF REGISTRATION TABLE   │
        │  INCLUDING SESSION IDENTIFIER,  │
        │  GAME IDENTIFIER                │
        │  AND PLAYER IDENTIFIER(S)  706  │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  SET SESSION STATUS FIELD TO    │
        │  "READY"                        │
        │                           707   │
        └─────────────────────────────────┘
                        │
                        ▼
        ┌─────────────────────────────────┐
        │  OUTPUT GAME REQUIREMENTS TO    │
        │  REGISTERED PLAYERS             │
        │                           708   │
        └─────────────────────────────────┘
```

FIG. 7

```
            ┌─────────────────────────┐
            │   BEGIN SECONDARY GAME   │
            │     CONTROL PROCESS      │
            └─────────────────────────┘
                         │
                         ▼
        ┌─────────────────────────────────┐
        │  RECEIVE PLAYER IDENTIFIER,      │
        │  CLIENT IDENTIFIER AND OUTCOME   │
        │                           810    │
        └─────────────────────────────────┘
                         │
                         ▼
              ╱──────────────────╲
             ╱    IS PLAYER        ╲      NO      ┌─────────┐
            ╱  REGISTERED FOR A GAME ╲──────────▶│   END   │
            ╲  THAT MAY BE PLAYED AT ╱           └─────────┘
             ╲   THIS MACHINE?      ╱
              ╲      812           ╱
               ╲─────────────────╱
                         │ YES
                         ▼
        ┌─────────────────────────────────┐
        │     RETRIEVE REGISTRATION        │
        │  DATA FROM IDENTIFIED RECORD     │
        │   OF REGISTRATION TABLE    814   │
        └─────────────────────────────────┘
                         │
                         ▼
        ┌─────────────────────────────────┐
        │   RETRIEVE GAME DATA FROM        │
        │   APPROPRIATE RECORD OF          │
        │  SECONDARY GAME TABLE      816   │
        └─────────────────────────────────┘
                         │
                         ▼
              ╱──────────────────╲
             ╱      DOES           ╲      NO      ┌─────────┐
            ╱  CORRESPONDING RECORD  ╲──────────▶│   END   │
            ╲   STATUS = "READY"     ╱           └─────────┘
             ╲    OR "ACTIVE"?      ╱
              ╲      818           ╱
               ╲─────────────────╱
                         │ YES
                         ▼
                        ( A )
                     TO FIG. 8B
```

FIG. 8A

FROM FIG. 8A

(A)

SESSION STATUS
= "READY"?

NO

820

YES

SET SESSION STATUS TO "ACTIVE"

822

DOES NUMBER
OF PLAYERS = 1?

YES

824

NO

DOES GAME TYPE
= "COMPETE"?

NO

(B)

TO FIG. 8C

826

YES

(C)

TO FIG. 8F

FIG. 8B

FROM FIG. 8B

(B)

DOES
OUTCOME MATCH A GAME
REQUIREMENT?
828

NO

YES

RECORD OUTCOME IN GAME
OUTCOME TABLE   830

UPDATE TIME REMAINING AND
PLAYS REMAINING   832

ARE ALL GAME
REQUIREMENTS
MET?
834

YES

(D)

TO FIG. 8D

NO

IS GAME
TERMINATED BASED ON
TIME / PLAYS?
836

YES

(E)

TO FIG. 8E

NO

END

FIG. 8C

FROM FIG. 8C

(D)

```
┌─────────────────────────────────────┐
│       SET SESSION STATUS FIELD OF    │
│  CORRESPONDING RECORD  IN REGISTRATION│
│      TABLE TO "SUCCESS"         838   │
└─────────────────────────────────────┘
```

```
┌─────────────────────────────────────┐
│       RETRIEVE APPROPRIATE GAME       │
│    BONUS AMOUNT FROM SECONDARY        │
│         GAME TABLE              844    │
└─────────────────────────────────────┘
```

```
┌─────────────────────────────────────┐
│  STORE GAME BONUS AMOUNT IN BONUS DUE │
│    FIELD OF APPROPRIATE RECORD IN     │
│       REGISTRATION TABLE        846   │
└─────────────────────────────────────┘
```

```
┌─────────────────────────────────────┐
│  SIGNAL SLOT MACHINE TO COMMUNICATE   │
│   "WIN" MESSAGE TO ALL ONLINE PLAYERS │
│   CORRESPONDING TO SESSION ID   848   │
└─────────────────────────────────────┘
```

(END)

FIG. 8D

FROM FIG. 8C

(E)

SET SESSION STATUS FIELD OF
CORRESPONDING REGISTRATION
RECORD TO "FAIL"   850

SIGNAL GAMING DEVICE TO COMMUNICATE
TERMINATION OF THE SECONDARY GAME TO
ALL ONLINE PLAYERS CORRESPONDING
TO SESSION IDENTIFIER   854

END

FIG. 8E

FROM FIG. 8B

(C)

DOES
OUTCOME MATCH A GAME
REQUIREMENT?
_860_

NO

YES

RECORD OUTCOME IN GAME
OUTCOME TABLE   _862_

UPDATE TIME REMAINING AND
PLAYS REMAINING   _863_

ARE ALL
GAME REQUIREMENTS
MET?
_864_

YES → (F)

TO FIG. 8G

NO

IS GAME
TERMINATED BASED ON
TIME / PLAYS?
_866_

NO → END

YES

DETERMINE WHICH PLAYER(S) MOST
SUCCESSFULLY COMPLETED GAME
REQUIREMENTS   _868_

(F)

TO FIG. 8G

FIG. 8F

FROM FIG. 8F

(F)

SET SESSION STATUS FIELD
OF WINNING PLAYER RECORD(S)
TO "SUCCESS"   870

SET SESSION STATUS FIELD OF
ALL OTHER PLAYER RECORD(S)
TO "FAIL"   872

RETRIEVE APPROPRIATE GAME
BONUS AMOUNT FROM AVAILABLE
GAMES DATABASE   874

RECORD GAME BONUS AMOUNT IN BONUS
DUE FIELD OF PLAYER RECORD(S)
CORRESPONDING TO SESSION IDENTIFIER IN
WHICH SESSION STATUS = "SUCCESS" 876

SIGNAL SUCCESSFUL PLAYER'S
GAMING DEVICE TO COMMUNICATE
"WIN" MESSAGE TO PLAYER   878

SIGNAL ALL OTHER PLAYERS' GAMING
DEVICES TO COMMUNICATE "GAME OVER"
MESSAGE TO PLAYERS   880

END

FIG. 8G

```
                    ┌──────────────────────────────┐
                    │   BEGIN REDEMPTION PROCESS    │
                    └──────────────────────────────┘
                                  │
                                  ▼
                    ┌──────────────────────────────┐
                    │  RECEIVE REQUEST TO PROVIDE   │
                    │    GAME BONUS AMOUNT          │
                    │  INCLUDING PLAYER             │
                    │  IDENTIFIER AND SESSION       │
                    │  IDENTIFIER          910      │
                    └──────────────────────────────┘
                                  │
                                  ▼
                    ┌──────────────────────────────┐
                    │  RETRIEVE PLAYER'S RECORD     │
                    │  FROM REGISTRATION TABLE      │
                    │  BASED ON SESSION IDENTIFIER  │
                    │  AND PLAYER IDENTIFIER   912  │
                    └──────────────────────────────┘
                                  │
                                  ▼
```

SESSION STATUS = "SUCCESS"? 914

NO → DISPLAY "INVALID REQUEST" MESSAGE 916

YES

PROVIDE APPROPRIATE PAYOUT BASED ON BONUS DUE 918

SET BONUS DUE FIELD OF PLAYER RECORD IN ACTIVE GAMES DATABASE TO "$0" 920

SET SESSION STATUS TO "PAID" 922

END

FIG. 9

—1000

CHERRY PICKER
SCAVENGER HUNT SLOT MACHINE GAME

SESSION ID # 0319981549

SIMPLY COLLECT THREE SEPARATE
CHERRY-CHERRY-CHERRY OUTCOMES
ON EACH OF THE MACHINES LISTED
BELOW TO WIN AN ADDITIONAL
$15 BONUS PAYOUT!

✓

YELLOW MACHINE   ☐

BLUE MACHINE   ☐

RED MACHINE   ☐

YOU HAVE **45 MINUTES** TO FULFILL THE
GAME REQUIREMENTS NECESSARY TO
RECEIVE THE BONUS PAYOUT.
GAME PLAY BEGINS ONCE YOU INSERT
YOUR PLAYER TRACKING CARD INTO ANY
ONE OF THE MACHINES LISTED ABOVE.
GOOD LUCK!

FIG. 10

ELECTRONIC AMUSEMENT DEVICE OFFERING SECONDARY GAME OF CHANCE AND METHOD FOR OPERATING SAME

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to an electronic amusement device and more particularly to an electronic amusement device such as a slot machine having the ability to conduct a secondary game of chance.

2. Description of the Related Art

In 1997, electronic amusement devices installed in U.S. casinos (including slot machines, video poker machines and the like, hereinafter referred to as "slot machines" or "machines") generated greater than ten billion dollars of revenue. With individual machines typically earning between $50 and $150 per day, slot machines often account for well over 50% of a U.S. casino's overall profits. The net profit from slot machine play for a casino typically exceeds the profit from all other casino activities.

The comparatively high profitability of slot machines may be attributed to many factors. One such factor is that slot machines typically offer a higher house advantage than other casino games. Further, slot machines typically enable a faster rate of play than other casino games. Another factor contributing to the higher profitability of slot machines is that slot machines may be enjoyable to players of every skill level. In addition, slot machines attract a large number of players by offering a large potential payout in exchange for a comparatively small wager. Slot machines also attract players who are intimidated by table games or other casino activities that require prior training or skill.

Because slot machine profitability is directly proportional to the speed of play, it would be advantageous for casino operators to encourage faster play at slot machines. It would further be desirable for casino operators to encourage players to try certain different types of slot machines. Specifically, casino operators would benefit from increased play at under-utilized machines, such as newer slot machines, older slot machines or slot machines located in a remote portion of a casino. Casino operators would also benefit from increased play on slot machines having a high house advantage and slot machines having a high maximum wager amount.

Presently, slot machines provide players with simple, passive entertainment. Although some recent slot machines enable a player to achieve rewards for nontraditional events, these slot machines do not require a player to perform any additional activities during game play.

An example of such a slot machine is disclosed by U.S. Pat. No. 5,639,088 entitled "Multiple Events Award System" of Schneider et al. (hereinafter referred to as "the '088 patent"). In the '088 patent, Schneider et al. disclose a system that enables a player to receive a large award for receiving a set of winning combinations within a pre-selected number of rounds of play. The system includes a central controller and a plurality of gaming machines. The gaming machines provide signals to the central controller representing a number of winning combinations and a number of rounds played. The central controller determines whether a player has received a set of winning combinations.

While the '088 patent enables a large award to be offered to a player, it fails to address certain problems with the prior art. Specifically, the '088 patent fails to encourage players to try certain different types of slot machines. The '088 patent also fails to increase play at under-utilized machines, such as older slot machines or slot machines located in a remote portion of a casino. The '088 patent further fails to direct players to slot machines having a high house advantage or slot machines having a high maximum wager amount.

Accordingly, it would be advantageous to provide a method and apparatus that encourages slot machine players to be directed to specific slot machines in a casino. Thus, it would be desirable to provide a method and apparatus which directs player traffic toward slot machines preferred by the casino, such as newer machines, highly profitable machines and otherwise under-utilized machines. It would further be desirable to provide a method and apparatus that encourages the active participation of a slot machine player, thus enhancing the gaming experience. Such a machine could result in significantly enhanced revenues for casino operators by attracting players who were previously uninterested in conventional slot play, while providing more enjoyable play for casino patrons.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a method and apparatus for conducting a secondary game of chance to be played using multiple gaming devices. An advantage of the present invention is that it directs players to utilize slot machines designated by a casino operator. A further advantage of the present invention is that it encourages a wider variety of players to utilize slot machines.

In accordance with a first aspect of the present invention, an electronic amusement device and method is disclosed for directing a computing device to register a player to play a secondary game of chance at a slot machine configured to conduct a primary game of chance and the secondary game of chance. The method includes the step of receiving a request to register the player to play the secondary game of chance. The request to register includes a player identifier. The method also includes the step of determining a secondary game identifier corresponding to the secondary game of chance. The secondary game identifier is associated with a set of client identifiers on which the secondary game of chance may be played. The method further includes the step of storing the player identifier, the secondary game identifier and the set of client identifiers, thereby registering the player for the secondary game of chance. The disclosed server operating in conjunction with a registration kiosk implements the steps of the described method.

In accordance with a second aspect of the present invention, an electronic amusement device and method is disclosed for directing a computing device to conduct a secondary game of chance at a client slot machine. The method includes the steps of receiving a player identifier corresponding to a player, and retrieving player data, including a secondary game type and a secondary game status. The secondary game type corresponds to game requirements for determining a winner of the secondary game of chance, and further corresponds to a set of eligible client identifiers. The method also includes the steps of determining a client identifier corresponding to the client slot machine, and determining whether the client identifier is associated with one of the eligible client identifiers. The method further includes the step of determining an outcome. The outcome is analyzed based on the game requirements of the secondary game. If all of the game requirements have been satisfied, the secondary game status is updated to reflect completion of the secondary game.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects, features and advantages of the invention will be understood from a consideration of the following description of the invention, in which:

FIG. 1 is a block diagram illustrating a system for implementing the present invention;

FIG. 2 is a block diagram of a slot server constructed in accordance with the present invention;

FIG. 3 is a block diagram of a client gaming device constructed in accordance with the present invention;

FIG. 4 is a block diagram of a registration kiosk constructed in accordance with the present invention;

FIGS. 5A–5B together comprise a table showing components of the secondary game definition table of FIG. 2;

FIGS. 6A–6C illustrate exemplary portions of the registration table and game outcome table of FIG. 2;

FIG. 7 is a flowchart illustrating a method for registering a player to play a secondary game of chance in accordance with the present invention;

FIGS. 8A–8G together comprise a flowchart illustrating a method for a slot server to control a secondary game of chance in accordance with the present invention;

FIG. 9 is a flowchart illustrating a method for redeeming winnings awarded during a secondary game of chance; and

FIG. 10 is a plan view of a game receipt generated in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

By directing slot players to specific slot machines or types of slot machines, casino operators can more effectively utilize the valuable floor space of a casino. By encouraging players to move to higher profit machines or encouraging an idle player to play any machine, casinos can achieve a higher profit per machine. Directing slot players to particular machines also benefits players by enhancing or expanding their gaming experiences. Many players are unfamiliar with certain machines or don't normally frequent areas of a casino housing particular slot machines. Encouraging a player to play a slot machine offering secondary rewards makes the player's experience more exciting.

The present invention is directed to a method and apparatus for registering and conducting a secondary game of chance on multiple client slot machines of a slot machine network. The preferred embodiment will be described with reference to secondary games of chance that are modeled after a scavenger hunt game. It should be understood that the present invention will function equally well with many other types of secondary games of chance, and that the use of scavenger hunt games is merely one exemplary embodiment.

Definitions

As used herein, the following terms will have the corresponding definitions:

Slot Machine: Any electronic amusement or gaming device typically installed in a gambling establishment, including slot machines, video poker machines and video blackjack machines.

Outcome: A set of one or more game elements, such as slot reel symbols or playing cards, used to determine a single game result.

Completion Parameter: A parameter or rule defining a completion condition, such as a set of game requirements defining successful completion of a game or a time limit defining unsuccessful completion of a game.

Session: A single instance of a secondary game of chance.

Client Identifier: An identifier representing a specific client slot machine or a family of client slot machines.

Apparatus Architecture

The apparatus architecture of an exemplary embodiment of the present invention will now be discussed with reference to FIGS. 1–4. Referring to FIG. 1, there is shown a block diagram of a slot network 100. Network 100 includes a slot machine server 200 (hereinafter referred to as server) that is linked to and communicates with registration kiosk 400 and client gaming devices or slot machines 300, 352 and 354. Although three client gaming devices 300, 352 and 354 are shown, a person of ordinary skill in the art will appreciate that any number of client gaming devices could be linked to and in communication with server 200.

In the exemplary embodiment, a player utilizes kiosk 400 to register to play a secondary game of chance, such as a scavenger hunt game. Preferably, upon receiving a request from the player, kiosk 400 provides the player with a list of available secondary games for which the player may register. Kiosk 400 retrieves data regarding available games from server 200 and presents the data to the player. The available game data may be presented in any conventional manner including via a menu on a touchscreen. If a player wishes to register for an available game, kiosk 400 collects registration information supplied by the player and provides the player with a game identifier and directions to at least one client gaming device on which the player may play the secondary game of chance. Registration information might include a player tracking card with a corresponding player identifier, or it may include player-specified information such as his name, address, and credit card number. Kiosk 400 may collect registration information from the player in a number of ways, including via a player-tracking card, keyboard interface, scanner or pointing device.

The player utilizes a client gaming device, such as client gaming device 300, 352 or 354, to play a primary game of chance offered by the client gaming device as well as the secondary game of chance for which the player is registered. Slot server 200 communicates with kiosk 400 and client gaming devices 300, 352 and 354 to manage the secondary game of chance.

Referring now to FIG. 2, the architecture of slot server 200 is illustrated. In addition to conventional server components, such as random access memory (not shown) and read only memory (not shown), slot server 200 includes a processor 210 linked to a clock 212, a storage device 214 and a communication port 220. Communication port 220 enables server 200 to communicate with registration kiosk 400 and client gaming devices 300, 352 and 354. Storage device 214 comprises an appropriate combination of magnetic and optical memory, such as disk drive memory, and semiconductor memory such as random access memory and read only memory, and contains program 216 for controlling server 200 in accordance with the present invention. Among other functions, program 216 includes instructions for registering a player for a secondary game of chance via kiosk 400, instructions for managing and controlling a secondary game of chance played at a client gaming device, and instructions for dispensing an award for successfully completing a secondary game of chance. Storage device 214 further includes relevant data, including player table 218, secondary game table 500, registration table 600 and game outcome table 700. The data stored by storage device 214 are described in more detail with reference to FIGS. 5–7. Although the data of the exemplary embodiment is stored at server 200, in alternate embodiments, the data may be

distributed among server 200, kiosk 400 and client gaming devices 300, 352 and 354.

Referring now to FIG. 3, the architecture of slot machine or client machine 300 is illustrated. Slot machine 300, which is substantially similar to slot machines 352 and 354, is controlled by processor 302 and communicates with slot server 200 via communication port 348. Processor 302 is connected to storage device 304 which stores program instructions and data for operating slot machine 300 in accordance with the present invention, including program 306, payout table 310 and probability table 312. Program 306 includes instructions for conducting the primary game of chance and instructions for conducting the secondary game of chance. Further connected to processor 302 are a player card tracking device 338, a random number generator 322, a reel controller 324, three reels 326, 328 and 330, a hopper controller 332 and associated hopper 334 and a currency acceptor 320.

As illustrated, slot machine 300 generally comprises conventional components, with the exception of the program instructions and data stored in storage device 304. For purposes of better illustrating the invention, standard components, well known to those skilled in the art, are described only briefly. Although the present embodiment of the invention is described as implemented with physical components, the invention applies equally well to and includes software embodiments such as would be implemented on the Internet and other computer data networks.

Referring again to processor 302, the slot machine 300 comprises one of many well known processing units, for example a Pentium class processor manufactured by Intel Corp. Data storage device 304 comprises an appropriate combination of magnetic and optical memory, such as disk drive memory, and semiconductor memory such as random access memory and read only memory. In addition to the program instructions and data shown in FIG. 3, storage device 304 stores appropriate operating system and control software (not shown), functional to operate gaming device 300 in the manner described below. Random number generator 322 comprises one of many well known random or pseudo-random number generators suitable for use in a gaming device.

Currency acceptor 320 is operative to receive one or more coins or bills, and to transmit an appropriate value signal to processor 302. Hopper controller 332, and hopper 334 connected thereto, are operative under the control of processor 302 to dispense coins to a player. Starting controller 350 comprises a player-operated device such as a handle or button for initiating the play of a game.

Player card tracking device 338 comprises a player tracking interface including a card reader 346 for receiving a player tracking card, a display 344 for communicating messages to the player, and a keypad 342 for receiving player input such as a player identifier. In addition to storing the player identifier, the player tracking card could be configured to store outcomes generated by slot machine 300 and other play related data, therefore obviating the need for registration table 600 and outcome table 700.

Referring now to FIG. 4, the architecture of registration kiosk 400 is illustrated. Kiosk 400 enables a player to register for a secondary game of chance and, in one embodiment, redeem winnings accumulated during the secondary game of chance. Kiosk 400 is controlled by processor 410 and communicates with slot server 200 via communication port 412. Processor 410 is connected to storage device 428 that stores, among other things, program 430. Program 430 includes instructions enabling a player to

register for a secondary game and instructions enabling a player to redeem bonus amounts awarded for successful completion of a secondary game.

Kiosk 400 includes input devices, such as currency acceptor 414 for receiving currency from the player, card reader 418 for reading a player tracking card, and input device 420 for receiving input from the player. Currency acceptor 414 enables kiosk 400 to accept a registration fee for the secondary game of chance, or pre-payment for the primary game of chance, in accordance with alternate embodiments of the present invention.

Kiosk 400 further includes output devices, including display 416 for displaying messages to the player, printer 422 and hopper controller 424 for dispensing currency via associated hopper 426. Printer 422 enables kiosk 400 to print receipts, reports, game instructions and coupons redeemable for free games, dinner, merchandise or cash. Hopper controller 424 enables kiosk 400 to dispense payment for awards earned for successful completion of the secondary game of chance.

Data Tables

Referring now to FIG. 5, there are illustrated six representative records of an exemplary secondary game table 500. Each record of secondary game table 500 defines the parameters of a secondary game that may be played, in addition to a primary game, at a client gaming device. Each record of secondary game table 500 includes a game identifier 510 that uniquely identifies the record and corresponds to the secondary game of chance that the record represents.

Each record of secondary game table 500 further includes field 512 that stores data representing the number of players required to play the associated secondary game. Game type field 514 stores data representing the type of secondary game. For example, record 550, having a game type of "SOLO," represents a single player game in which a single player must fulfill game requirements 524 to achieve a bonus. Record 560, having a game type of "COMBINATION," represents a multiple-player game in which the players cooperate as a team to collect outcomes which fulfill game requirements 524 to achieve a bonus. Record 570, having a game type of "COMPETITION," represents a multiple-player game in which the players compete to be the first to collect outcomes which fulfill the game requirements to achieve a bonus. Secondary games may be defined so that game requiring multiple players have more requirements than single player games, and the greater the number of required players, the longer or more complex the list of requirements.

Each record of secondary game table 500 further includes a number of parameters which define a secondary game, including a game time limit field 516, game play limit field 518, game bonus amount field 520 and client identifiers field 522. Game time limit field 516 stores a time value that represents the maximum amount of time within which a player must collect the required game elements or outcomes to qualify for an award. Game play limit field 518 stores a value representing the maximum number of plays within which a player must collect the required slot symbols to qualify for an award. A single play might represent the pull of a traditional slot machine handle, completion of a final hand in video poker or completion of one hand of blackjack. Game bonus amount field 520 stores a monetary value that a player is awarded for completing the game requirements. Of course the bonus could be paid to a single player, divided among a group of players, or provided in full to each of a group of players. In addition to a bonus, a player successfully completing a secondary game of chance could win the

right to play another game. This would enable tournament play among secondary game players.

Client identifiers field 522 stores the client identifiers of the client gaming devices on which the secondary game may be played. In the preferred embodiment, each client identifier stored in field 522 represents a family or type of slot machine, however, embodiments in which each client identifier stored in field 522 represents a specific slot machine are also envisioned. Of course, client identifier field 522 may store data representing a rule for identifying one or more client gaming devices. For example, in an embodiment in which descriptive client identifiers are assigned to client gaming devices (e.g. slot machine identifiers begin with "SM" and video poker identifiers begin with "VP"), client identifier field 522 may store a rule limiting eligible client gaming devices to those beginning with "SM."

In addition, each record of secondary game table 500 includes at least one field defining the requirements of the corresponding secondary game. As illustrated, game requirements field 524 stores a definition of the requirements of each secondary game. Of course, one of ordinary skill will realize that game requirements field 524 is merely exemplary, and in practice, the requirements of each secondary game may be defined by multiple fields containing defining parameters. Alternatively, the requirements of each secondary game may be implemented through program instructions.

As illustrated, the game requirements require the player to collect specific outcomes from designated machines. The game requirements could easily require an order in which the player must collect the outcomes. For example, the player might have to complete requirements on lower denomination machines before progressing to higher denominations. An order of individual machines could also be specified, allowing the casino to distribute play over a fixed number of machines by specifying a different starting machine for each secondary game registered for that particular set of client machines.

Game requirements could also specify a minimum number of client machines on which a player must play in order to successfully complete the secondary game. By specifying a time window for play, the casino could encourage more play during off-peak times. For example, secondary games might only be available for play Monday through Thursday from 10:00 AM to 2:00 PM.

It is also envisioned that the outcomes designated by the secondary game requirements may consist entirely of game elements or outcomes that do not constitute winning outcomes in the primary game. In other words, in order to achieve an outcome required by the secondary game, a player must forego a winning outcome in the primary game of chance, and vice versa.

Much of the information stored in secondary game table 500 is presented to a player at kiosk 400 during a registration process. Kiosk 400 receives a request to register the player for a secondary game, and displays the available games, as defined in secondary game table 500, to the player for selection. Upon receiving a game selection from the player, kiosk 400 may direct the player to the associated client gaming devices by displaying, printing or otherwise providing the player with a map of the casino floor, with the associated machines explicitly identified on the map. For ease of explanation, the present invention is disclosed with respect to an embodiment in which the casino color-codes certain types of machines in order to make them easy to recognize. This could be done by placing a placard or other visible sign on each associated machine, clearly marking its

color code designation. In embodiments in which a particular order of client machines is required in order to satisfy the secondary game requirements, the map might indicate a preferred or required route, perhaps through the use of arrows or shading. In this embodiment, a player that is required to receive a cherry-cherry-cherry outcome on a blue machine simply has to look around for a machine with the blue placard on it. Server 200 recognizes an eligible client based on a client identifier received from the client machine. Server 200 may use the client identifier directly to determine whether a player may use the client device to play the secondary game. Alternatively, server 200 may use the client identifier to reference a client eligibility table (not shown) to indirectly determine whether a player may use the client device to play the secondary game.

In addition to the type of games illustrated, many other types of games could be implemented. For example, the player may be required to receive three cherry-cherry-cherry outcomes before he receives five lemon-lemon-lemon outcomes. As described with reference to FIGS. 6A–6C, server 200 would keep track of the player's cherry-cherry-cherry outcomes and lemon-lemon-lemon outcomes. If the player collects three cherry-cherry-cherry outcomes before he collects five lemon-lemon-lemon outcomes, the player is successful and is eligible to receive the bonus associated with the game. If, however, the player receives five lemon-lemon-lemon outcomes before receiving three cherry-cherry-cherry outcomes, the player is unsuccessful, and is not eligible to receive the associated bonus.

Although the records of secondary game table 500 are preferably populated by experienced casino personnel who understand the subtleties of maximizing the efficient use of the casino floor, server 200 may be programmed to automatically generate records for secondary game table 500 based on historical data collected from client devices 300, 352 and 354. For example, as one of ordinary skill will appreciate, historical coin-in data may be collected from client devices 300, 352 and 354. Server 200 could be programmed to analyze the coin-in data and to rank the client devices. In order to promote play on the lower ranked client devices, server 200 might be programmed to generate a record in secondary game table 500 to provide a secondary game directed to the lower ranked client devices.

Referring now to FIG. 6A, an exemplary record 650 from registration table 600 and two corresponding records 750 and 752 from game outcome table 700 are shown. Each record of registration table 600 represents a session of a secondary game for which a player is registered. Registration table 600 includes a session identifier 610 uniquely identifying the registration record. In the preferred embodiment, session identifier 610 is generated for the session by the server based on the date and time of registration. Registration table 600 further includes a game identifier 612 and a player identifier 614 for identifying the secondary game and the player associated with the session. Game identifier 612 is populated with the data from game identifier field 510 of the appropriate record from secondary game table 500, and player identifier 614 is populated with the player identifier of the registered player. Player identifier 614 is preferably generated and assigned by the casino, but could take the form of a personal identification number (PIN) selected by player. Player identifier 614 is captured at kiosk 400 during a registration process. Player identifier 614 may be read from a player tracking card inserted into card reader 418, or may be provided via input device 420.

Registration table 600 further includes fields for tracking the progress of the player toward completion of the game

requirements. Time remaining field 616 stores a time value representing the amount of time available for the player to complete the game requirements. The time available could be tracked individually for each player, or for a team of players. In such a team embodiment, the time remaining for each team member begins counting down with the first player inserting his player tracking card. For an individual player, the amount of time remaining could be frozen each time the player ended a particular series of plays by removing his player tracking card, with the time starting to count down only after the card was again inserted. Alternatively, time remaining field 616 could begin to count down immediately following registration by the player, encouraging the player to immediately begin playing the secondary game. Plays remaining field 618 stores a value representing the number of plays available for the player to complete the game requirements. Initially, time remaining field 616 and plays remaining field 618 are populated with data from game time limit field 516 and game play limit field 518 of the appropriate record from secondary game table 500, respectively.

Session status field 620 stores a code representing a status of the corresponding session. Examples of valid session status codes are listed in Table I, below. In the preferred embodiment, a session is successfully completed when the player satisfies the corresponding game requirements. A session is unsuccessfully completed when the player has not satisfied the game requirements and there is no time or plays remaining. Of course, various other session completion parameters are possible. Bonus due field 622 stores a monetary value representing a bonus due to the player for successfully completing the session. Although an unsuccessfully completed session will result in a bonus due amount of zero, it should be understood that varying degrees of success may be defined by the game requirements, resulting in varying bonus due amounts.

TABLE I

Session Status Code	Session Status Definition
READY	A player has been registered to play a session of a game.
ACTIVE	A session of a game is in progress.
SUCCESS	A player successfully fulfilled the game requirements for the session.
FAIL	A player failed to successfully fulfill the game requirements for the session.
PAID	A bonus for successfully completing the game requirements has been paid to the player.

Each record of game outcome table 700 represents a game outcome that contributes to the successful completion of the session. Game outcome table 700 includes a session identifier 710 that identifies the session corresponding to the game outcome. Game outcome table 700 also includes player identifier 711, outcome 714 and client identifier 712 for respectively documenting a player, an outcome fulfilling a game requirement (e.g. a reel symbol, set of reel symbols or card values) and the client gaming device that generated the outcome. Game outcome table 700 preferably includes a timestamp field 716 for storing the date and time the associated outcome was generated. Timestamp field 716 may be used to audit winning games. In alternate embodiments, timestamp field 716 may be used to cause outcomes to expire after a pre-specified period of time or number of plays.

The records shown in FIG. 6A describe a session of game ABC being played by a player having player identifier

24681012. To successfully complete game ABC, the player must receive 100 lemons during the session of up to 225 plays as defined by record 550 of secondary game table 500, previously described with reference to FIG. 5. According to session status field 618 of record 650, the session is "ACTIVE," or in progress. Although the player was initially allotted 225 plays to complete the game requirements of game ABC, plays remaining field 624 indicates that the player has completed 120 plays and has 105 plays remaining to complete the game requirements. As shown by records 750 and 752 of game outcome table 700, player 24681012 has received two lemons while playing slot machines having client identifiers SM-1002003 and SM 1001098.

The records shown in FIG. 6B describe a session of game DEF being played by two players having player identifiers 36917154 and 48129003. To successfully complete game DEF, the players must cooperate to achieve the game requirements defined by record 560 of secondary game table 500, previously described with reference to FIG. 5. According to session status field 620 of record 652, the session has been registered, but play has not begun. The initial allotment of sixty minutes is stored in time remaining field 616 of records 652 and 654. Because the secondary game has not started, there are no relevant records in game outcome table 700 relating to session 0327981148.

The records shown in FIG. 6C describe a completed session of game EFG, a competitive game between two players having player identifiers 35791130 and 24483696. To successfully complete game EFG, a player must be the first to collect four outcomes: (i) a pair of jacks or better on any video poker machine, (ii) a cherry-cherry-cherry outcome on any three reel slot machine, (iii) three-of-a-kind on any video poker machine, and (iv) orange-orange-orange on any blue machine while wagering the maximum amount, as illustrated by record 570 of FIG. 5. As shown by records 754, 756, 758 and 760, the player having player identifier 35791130 successfully completed the game requirements.

Description of the Operation

Having thus described the architecture and components of the slot network of the present embodiment, the operation of the apparatus will now be described in greater detail with reference to FIGS. 7–9. FIG. 7 is a flowchart illustrating an exemplary registration process; FIGS. 8A–8G together comprise a flowchart illustrating an exemplary secondary game control process; and FIG. 9 is a flowchart illustrating an exemplary redemption process. These flowcharts describe a preferred embodiment in which server 200 facilitates registration and redemption processes via kiosk 400, and game play via client gaming device 300. It should be understood, however, that the processes do not strictly require the described client-server architecture. For example, a gaming device providing a single player secondary game of chance could conduct registration, game play and redemption processes at a single unit.

Referring now to FIG. 7, an exemplary registration process is illustrated in the form of a flowchart. The registration process enables a player to select a secondary game in which to participate. The registration process preferably interfaces with the player on the casino floor via kiosk 400, but may be accomplished by a client gaming device 300.

At block 701, processor 410 receives a request to register a player for a secondary game of chance and forwards the request to slot server 200 via communication port 412. At block 702, slot server 200 retrieves data describing the available games from secondary game table 500 and transmits the data to kiosk 400 where processor 410 causes the data to be communicated to the player via display 416.

11

Kiosk **400** receives the game selection of the player at block **703**. The player may indicate a game selection using a touch screen or by providing a game identifier from a list of games provided at block **702**. At block **704**, processor **410** determines at least one player identifier, and at block **705** processor **410** determines a session identifier. The number of player identifiers determined at block **704** is based on the number of players required to play the selected game. The player identifiers may be determined in a number of ways including receiving a player identifier from a player tracking card inserted into card reader **418**, receiving a player-selected PIN, or generating a unique player identifier at the time of registration.

If the selected game requires multiple player identifiers, the registration process instructions of program **430** will enable multiple identifiers to be collected and verified. If the appropriate number of player identifiers are not received, kiosk **400** may display a message indicating that the identifiers were not received or were invalid.

For example, if the player requests to register for a three player game, he must provide two player identifiers in addition to his own. If the player fails to provide the correct number of player identifiers, his request to register for the secondary game will be rejected. In an alternate embodiment, players who do not have player tracking cards (or previously assigned player identifiers) but still wish to participate in a secondary game of chance may be provided with a PIN that will serve as their identifier for the duration of the game. The PIN may be either player-selected or generated by the system. Each player registered for a game will receive a unique PIN which may be input by the player via keypad **342** prior to playing a secondary game.

At block **706**, processor **410** transmits registration data including the session identifier, game identifier and player identifier(s) to server **200** which creates a new record in registration table **600** and stores the registration data in the new record. At block **707**, server processor **202** sets session status field **620** to "READY." The requirements of the selected game are output to the player at block **708**. An illustration of such output is described with reference to FIG. **10**. The output could alternatively take the form of a map showing the player the locations of various client gaming machines associated with the particular secondary game, printed in real-time or developed in advance as a brochure with casino promotional materials.

Instead of providing detail regarding the identity of each client machine, the registration process could identify a first client machine with subsequent client machines identified during play of the secondary game. Thus, the player does not know where he is heading in advance, receiving the identification of his next client machine from display **344** of client machine **300**. The player might also not be informed of the exact requirements of the secondary game until it was underway. For example, the player might achieve a required outcome of lemon-lemon-lemon and then receive identification of the next game requirement from display **344** such as bell-bell-bell.

Referring now to FIGS. **8A–8G**, an exemplary secondary game control process is illustrated in the form of a flowchart. The illustrated secondary control process is performed by slot server **200**, for every outcome generated by a client gaming device, in accordance with program **216**.

Generally, if the player operating the client gaming device is registered for a game, server **200** will determine the type of game based on the game identifier, and direct the gaming device to execute the appropriate game play steps. If the player is not registered for a game, server **200** will not execute any steps to control a secondary game of chance. Once server **200** determines that the player is registered for a secondary game, it will update the time remaining and/or plays remaining while the player's player tracking card is in the gaming device, and store any of the player's outcomes that satisfy a game requirement. Server **200** will further store the client identifier of the client gaming device on which the outcome was generated, as well as the player identifier of the player playing the game. The record of the game outcome table is linked to the player's record of the registration table through a session identifier and the player's player identifier. In an alternate embodiment, server **200** could be programmed to continue to update the time remaining field of secondary games in progress so that if a player moves from one machine to another, the time remaining will continue to decrease. In the alternate embodiment, server **200** would continually update the time remaining from the time the player begins playing the secondary game until the completion of the game, regardless of whether the player removes the player tracking card from the gaming device.

In a multi-player team embodiment (either combined or competitive), the server keeps track of each player's results in the manner described above. In the competitive multi-player team embodiment, server **200** monitors each player's outcomes in substantially real time, and terminates the game once a player fulfills all of the game requirements. In an alternate embodiment, server **200** could track the time each player's outcome was obtained, in order to determine which player was the first to fulfill all of the requirements of the game.

At block **810**, processor **210** receives a player identifier, client identifier and outcome from client gaming device **300**. The player identifier, client identifier and outcome are transmitted by client gaming device **300** after a play of the machine. The transmission is performed in accordance with program **306**.

At block **812**, processor **210** determines whether a record corresponding to the received player identifier is registered for a secondary game of chance and whether the client identifier is among the eligible client identifiers associated with the secondary game. If a corresponding record does not exist, no further processing is required to control a secondary game of chance because the player associated with the received player identifier is not registered for a secondary game of chance that may be played at client gaming device **300**. If registration record corresponding to the received player identifier and client identifier exists, secondary game processing continues.

At block **814**, processor **210** retrieves registration data from the identified record of registration table **600**. Processor **210** retrieves game data from an appropriate record of secondary game table **500**, as shown by block **816**. The appropriate secondary game table record is determined based on the contents of game identifier field **612** retrieved at block **814**. At decision block **818**, processor **210** determines whether the retrieved session status field **620** contains "READY" or "ACTIVE." If the retrieved registration record does not have a session status of either "READY" or "ACTIVE," the registered game is complete, and processing terminates. Otherwise, at block **820** of FIG. **8B** processor **210** determines whether the session status field contains "READY." If it does, the session status is set to "ACTIVE" at block **822** to indicate that the session is in progress. At blocks **824** and **826**, process flow is directed based on whether the secondary game is a single player game, a multi-player competitive game or a multi-player combination game.

If the secondary game is a single player game or a multi-player cooperative game, process flow is directed to block 828 of FIG. 8C. At block 828, processor 210 determines whether the outcome, or any portion of the outcome, matches a game requirement. If the outcome fulfills a game requirement, the outcome is recorded in game outcome table 700, as shown by block 830. At block 832, the time remaining and plays remaining fields of registration table 600 are updated. Processor 210 then determines whether all of the game requirements have been met at decision block 834. If all of the game requirements have been satisfied, processing is directed to block 838 of FIG. 8D, enabling the player to be informed of his success in substantially real time. Otherwise, processor 210 determines whether the game should be terminated due to the lack of remaining time or remaining plays. According to decision block 836, if the game should be terminated, processing is directed to block 850. Otherwise, the secondary game control process terminates. It should be noted that the process flow may be altered to enable the player to be informed of his success or failure at the end of the allotted time/plays without deviating from the spirit and scope of the present invention.

Referring now to FIG. 8D, there are illustrated the steps processor 210 executes if a player wins a single player game or cooperative multi-player game. At block 838, processor 210 is directed to set the session status field 620 of the corresponding record of registration table 600 to "SUCCESS." Processor 210 then retrieves game bonus amount 520 from the appropriate record of secondary game table 500 and stores the retrieved amount in the bonus due field 622 of the corresponding record of registration table 600, as shown by blocks 844 and 846. In an alternate embodiment, instead of analyzing whether a player has satisfied the game requirement in real time, server 200 may be programmed to determine success or failure of a player only after expiration of the time remaining or number of plays remaining.

At block 848, processor 210 causes a signal to be transmitted to all client gaming devices on which a player associated with the session identifier is playing. The signal directs the client gaming devices to display a message indicating that the player has successfully completed the game requirements. In a multi-player game, all players are to be notified of the successful completion of the game requirements. If server 200 is unable to direct a message to a player because the player is not presently operating a gaming device, the message may be queued for delivery as soon as the player resumes operation of a client gaming device.

Referring now to FIG. 8E, there are illustrated the steps processor 210 executes if a player fails to complete the game requirements within the allotted time or number of plays. At block 850, processor 210 sets session status field 620 of the corresponding record of registration table 600 to "FAIL." At block 854, processor 210 causes a signal to be transmitted to every client gaming device on which a player associated with the session identifier is playing. The signal directs the client gaming devices to display a message indicating that the player secondary game has ended unsuccessfully.

Referring now to FIGS. 8F and 8G, there are illustrated the steps processor 210 executes if the corresponding session identifier is associated with a competitive multi-player game. At decision block 860, processor 210 determines whether the outcome, or any portion of the outcome, matches a game requirement. If the outcome fulfills a game requirement, the outcome is recorded in game outcome table 700, as shown by block 862. At block 863, the time remaining and plays remaining fields of registration table 600 are updated. Processor 210 then determines whether all of the game requirements have been met at decision block 864. If all of the game requirements have been satisfied, processing is directed to block 870 of FIG. 8G. Otherwise, processor 210 determines whether the game should be terminated due to the lack of remaining time or remaining plays. According to decision block 866, if the game should be terminated, processing is directed to block 868. Otherwise, the secondary game control process terminates.

At block 868, processor 210 determines which players most successfully completed the game requirements. Pursuant to blocks 870 and 872, respectively, session status field 620 of the corresponding record of registration table 600 is updated to "SUCCESS" for all winning players and "FAIL" for all other players. Although not shown by secondary game table 500, multi-player games may be defined in which a bonus may be awarded for multiple levels of achievement, such as first place, second place and third place. Processor 210 then retrieves game bonus amount 520 from the appropriate record of secondary game table 500 and stores the retrieved amount in the bonus due field 622 of the corresponding records of registration table 600, as shown by blocks 874 and 876. At block 878, processor 210 causes a signal to be transmitted to all client gaming devices on which a winning player associated with the session identifier is playing. The signal directs the client gaming devices to display a message indicating that the player has successfully completed the game requirements. At block 880, processor 210 causes a signal to be transmitted to all client gaming devices on which a losing player associated with the session identifier is playing. The signal directs the client gaming devices to display a message indicating that the player has failed to successfully complete the game requirements.

FIGS. 8F and 8G assume that the secondary game incorporates a time/plays limit, and that if the time/plays limit is reached the player having met the most game requirements is deemed to be the winner. Of course, the secondary game process could be easily modified to require a winner to complete all of the game requirements by eliminating block 868 from the process of FIG. 8F and setting session status field 620 of all of the participating players to "Fail" at block 872 of FIG. 8G.

Referring now to FIG. 9, there are illustrated the steps of an exemplary redemption process. The redemption process is typically initiated by a player who has registered for and successfully completed the game requirements of a secondary game of chance. The redemption process is typically performed by server 200 operating in conjunction with kiosk 400.

At block 910, server 200 receives a request to provide a game bonus amount to a player. The request is received via kiosk 400 and includes a player identifier identifying the requesting player and a session identifier identifying the session during which the player achieved a game bonus. Server 200 retrieves the record from registration table 600 which corresponds to the received session identifier and player identifier, as shown by block 912.

Server 200 determines whether the player successfully completed the session by comparing session status field 620 to "SUCCESS" at block 914. If the identified session does not have an associated status of "SUCCESS," server 200 prompts kiosk 400 to display a message indicating that the request is invalid, as shown by block 916, and the redemption process terminates. If the player successfully completed the session, server 200 causes the appropriate payout to be provided based on the contents of bonus due field 622, as shown at block 918. At blocks 920 and 922, respectively,

bonus due field **622** is zeroed and session status field **620** is updated to "PAID" to reflect payment of the bonus. The game bonus may be automatically dispensed by kiosk **400**, or manually dispensed by a casino attendant. In an alternate embodiment, the redemption process could be executed by the gaming device at which the final game requirement is completed, or executed by any client slot machine. It should be noted that a bonus award earned by a team may be dispensed in a number of ways in accordance with the present invention. For example, each team member may receive the entire bonus amount, each team member may receive an equal share of the bonus amount, or each player may receive a share of the bonus amount proportional to the game requirements achieved by the player.

Bonus payout amounts could be adjusted based on the time to successful completion, with shorter completion times associated with relatively larger bonus payout amounts. Bonus payout amounts could also be supplemented in order to encourage play at off-peak hours, such as during early morning hours. In order to encourage a faster rate of play, the casino could offer higher bonus payout amounts to those players maintaining a high average rate of play. For example, a player completing a secondary game with a seven hundred handle pull-per-hour average rate of play might receive 10% more than a player maintaining a rate of only five hundred handle pulls-per-hour for the same secondary game structure.

Referring now to FIG. **10**, there is illustrated an exemplary game receipt **1000** that may be provided to a registered player. Game receipt **1000** acts as proof of registration for a secondary game session and provides a convenient reminder to the player of the game requirements, time limit and bonus amount associated with the secondary game. Game receipt **1000** further directs the player to the appropriate gaming devices, and may provide instructions regarding redemption of an earned bonus. As shown, game receipt also includes check boxes enabling the player to track his progress toward successful completion of the game requirements.

While the best mode for carrying out the invention has been described in detail, those familiar with the art to which the invention relates will recognize various alternative designs and embodiments for practicing the invention. These alternative embodiments are within the scope of the present invention. Accordingly, the scope of the present invention embodies the scope of the claims appended hereto.

What is claimed is:

1. A method, comprising:

receiving a player identifier corresponding to a player playing a primary game at a gaming device;

determining secondary game completion parameters defining a secondary game, the secondary game completion parameters comprising:

at least a first gaming device and a second gaming device that is different from tie first gaming device, wherein the primary game is to be played by the player on the at least a first gaming device and a second gaming device that is different from the first gaming device;

a plurality of eligible gaming devices; and

a plurality of game requirements that the player must obtain on the at least a first gaming device and a second gaming device that is different from the first gaming device, wherein a game requirement comprises at least one of an outcome of the primary game and at least one element of an outcome of the primary game;

determining an outcome of the primary game; and

determining whether a game requirement has been obtained by the player by analyzing whether the outcome was obtained on one of the eligible gaming devices and whether the outcome satisfies at least one of the plurality of game requirements.

2. The method of claim 1, wherein the step of determining secondary game completion parameters comprises:

accessing secondary game completion parameters that are stored at a gaming device.

3. The method of claim 1, wherein the step of determining secondary game completion parameters comprises:

accessing secondary game completion parameters that are stored at the gaming device.

4. The method of claim 1, wherein the step of determining secondary game completion parameters comprises:

accessing secondary game completion parameters that are stored on a player tracking card.

5. The method of claim 1, wherein the secondary game completion parameters are associated with the player identifier.

6. The method of claim 5, wherein the player had previously registered for the secondary game.

7. The method of claim 1, wherein the secondary game completion parameters further comprise:

a time period within which the player may obtain the plurality of game requirements.

8. The method of claim 1, wherein the secondary game completion parameters further comprise:

a specified order in which the player has to play the eligible gaming devices.

9. The method of claim 1, wherein the secondary game completion parameters further comprise:

a specified order in which the player must obtain each of the plurality of game requirements.

10. The method of claim 1, wherein the secondary game completion parameters further comprise:

a requirement that each of the plurality of game requirements be obtained by the player at a specific corresponding one of the plurality of eligible gaming devices, respectively.

11. The method of claim 1, further comprising:

storing, in association with the player identifier, an indication that the at least one of the plurality of game requirements has been obtained by the player if it is determined that at least one of the plurality game requirements has been obtained by the player.

12. The method of claim 11, wherein the step of storing an indication that the at least one of the plurality of game requirements has been obtained by the player comprises:

storing, in association with the player identifier, an indication of the outcome if it is determined that the at least one of the plurality of game requirements has been obtained by the player.

13. The method of claim 11, wherein the step of storing an indication of the outcome comprises:

storing, in association with the player identifier, an indication of the at least one element of the outcome that satisfies the at least one of the plurality of game requirements.

14. The method of claim 11, wherein the indication that the at least one of the plurality of game requirements has been obtained by the player is stored at a server in communication with a gaming device.

15. The method of claim 11, wherein the indication that the at least one of the plurality of game requirements has been obtained by the player is stored at a gaming device.

16. The method of claim 11, wherein the indication that the at least one of the plurality of game requirements bas been obtained by the player is stored on a player tracking card that the player insert into a gaming device.

17. The method of claim 1, wherein the outcome does not qualify for an award in the primary game.

18. The method of claim 1, wherein the outcome qualifies for an award in the primary game.

19. The method of claim 1, further comprising:

outputting to the player an indication that at least one of the plurality of game requirements has been obtained by the player based on the step of determining whether a game requirement has been obtained by the player.

20. The method of claim 1, wherein the secondary game completion parameters further comprise:

a secondary game award to be provided to the player if the player obtains each of the plurality of game requirements and satisfies the secondary game completion parameters.

21. The method of claim 20, further comprising:

determining whether the player has obtained each of the plurality of game requirements and satisfied the secondary game completion parameters; and

causing the secondary game award to be provided to the player based on the step of determining.

22. The method of claim 1, further comprising:

outputting an indication of one of the plurality of game requirements that the player has not yet obtained to the player after the player obtains one of the plurality of game requirements.

23. The method of claim 1, further comprising:

outputting an indication of each of the game requirements to the player at one time.

24. A method, comprising:

receiving a player identifier corresponding to a player;

determining secondary game completion parameters defining a secondary game, the secondary game completion parameters comprising:

at least a first gaming device and a second device that is different from the first gaming device, wherein the primary game is to be played by the player on the at least a first gaming device and a second gaming device that is different from the first gaming device;

a plurality of eligible gaming devices;

a plurality of game requirements that the player must obtain on the at least a first gaming device and a second device that is different from the first gaming device, wherein a game requirement comprises at least one of an outcome of a primary game and at least one element of an outcome of the primary game; and

a secondary game award;

determining a plurality of outcomes obtained by the player while playing the primary game at a plurality of gaming devices; and

causing the secondary game award to be provided to the player if the plurality of outcomes obtained satisfy the secondary game completion parameters and the game requirements.

25. The method of claim 24, wherein the method is performed by at least one of (i) a server in communication with the plurality of gaming devices, (ii) one of the plurality of gaming devices, and (iii) a kiosk.

26. The method of claim 24, further comprising:

receiving, from the player, at least one of (i) a request for the secondary award and (ii) an indication that the

player has satisfied the secondary game completion parameters and obtained each of the plurality of game requirements.

27. The method of claim 24, wherein the step of determining the plurality of outcomes obtained by the player comprises:

accessing the plurality of outcomes obtained by the player, wherein the plurality of outcomes are stored on at least one of (i) a server in communication with the plurality of gaming devices, (ii) at least one of the plurality of gaming devices, and (iii) a player tracking card associated with the player.

28. The method of claim 24, wherein the secondary game completion parameters further comprise:

a time period within which the player must obtain the plurality of game requirements.

29. The method of claim 28, further comprising:

determining a beginning of the time period for the player; and

performing the method of claim 24 at an end of the time period.

30. The method of claim 24, wherein the step of causing the secondary game award to be provided to the player comprises at least one of (i) dispensing the award to the player at one of the plurality of gaming devices, (ii) dispensing the award to the player at a kiosk, and (iii) dispensing the award to the player via a casino attendant.

31. The method of claim 24, wherein the secondary game award comprises at least one of (i) a monetary payment, (ii) an amount of casino currency, and (iii) eligibility for another secondary game.

32. The method of claim 24, wherein the secondary game completion parameters further comprise:

one of a minimum and maximum number of plays of a primary game within which the player must obtain the game requirements.

33. The method of claim 24, wherein the player had previously registered for the secondary game.

34. A method, comprising:

receiving a first player identifier that identifies a first player;

receiving a second player identifier that identifies a second player;

determining secondary game completion parameters defining a secondary game, the secondary game completion parameters comprising:

at least a first gaming device and a second gaming device that is different from the first gaming device, wherein a primary game is to be played by the first player and the second player on the at least a first gaming device and a second gaming device that is different from the first gaming device;

a plurality of eligible gaming devices; and

a plurality of game requirements that the first player and the second player must obtain, wherein a game requirement comprises at least one of an outcome of the primary game and at least one element of an outcome of the primary game;

determining an outcome obtained by the first player while playing the primary game on a gaming device; and

determining whether a game requirement has been obtained by the first player by analyzing whether the gaming device is one of the eligible gaming devices and whether the outcome satisfies at least one of the plurality of game requirements.

35. The method of claim **34**, further comprising:

storing, in association with the first player identifier and the second player identifier, an indication that a game requirement has been obtained if the outcome obtained by the first player satisfies at least one of the plurality of game requirements and if the secondary game completion parameters have been satisfied.

36. The method of claim **34**, wherein the secondary game completion parameters further comprise:

a secondary game award to be provided to at least one of the first player and the second player if the first player and the second player obtain the game requirements and satisfy the secondary game completion parameters.

37. The method of claim **34**, further comprising:

determining that each of the plurality of game requirements has been obtained by the first player and the second player and that the secondary game completion parameters have been satisfied by the first player and the second player, thereby determining that the first player and the second player qualify for the secondary game award; and

causing the secondary game award to be provided to at least one of the first player and the second player.

38. The method of claim **37**, further comprising:

outputting an indication that the first player and the second player qualify for an award to at least one of the first player and the second player.

39. The method of claim **34**, wherein the secondary game completion parameters further comprise:

a requirement that the first player obtain a first subset of the plurality of game requirements; and

a requirement that the second player obtain a second subset of the plurality of game requirements.

40. The method of claim **34**, wherein the secondary game completion parameters further comprise:

a requirement that the first player play the primary game at a first subset of the eligible gaming devices; and

a requirement that the second player play the primary game at a second subset of the eligible gaming devices.

41. The method of claim **34**, further comprising:

outputting an indication to the second player that the first player has obtained at least one of the plurality of game requirements if the outcome obtained by the first player satisfies at least one of the plurality of game requirements and the secondary game completion parameters.

42. The method of claim **41**, wherein the step of outputting comprises outputting the indication via a display device of a gaming device at which the second player is playing.

43. The method of claim **34**, wherein the first player and the second player had previously registered for the secondary game.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO.   : 6,364,765 B1

Page 1 of 1

DATED   : April 2, 2002

INVENTOR(S)   : Jay S. Walker et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

<u>Title page,</u>
Item [75], insert -- **Robert R. Lech**, Norwalk, CT --

Signed and Sealed this

Fourth Day of February, 2003

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

United States Patent [19]

Nakamura

[11] Patent Number: 5,580,308

[45] Date of Patent: Dec. 3, 1996

[54] **ELECTRONIC BATTLE GAME PLAYING APPARATUS WITH FACIAL MONTAGE GENERATION**

[75] Inventor: **Toshiki Nakamura**, Tachikawa, Japan

[73] Assignee: **Casio Computer Co., Ltd.**, Tokyo, Japan

[21] Appl. No.: **272,420**

[22] Filed: **Jul. 8, 1994**

[30] **Foreign Application Priority Data**

Jul. 14, 1993 [JP] Japan 5-196962
Jul. 14, 1993 [JP] Japan 5-196963

[51] Int. Cl.^{6} A63F 9/24
[52] U.S. Cl. 463/7; 463/1
[58] Field of Search .... 273/433, 434, 273/437, 438, 85 G, 148 B, DIG. 28; 364/410; 434/270, 81, 185; 395/133, 155, 161, 152; 345/121, 122; 463/1, 7, 8

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,104,625	8/1978	Bristow et al.	273/DIG. 28
4,315,628	2/1982	Balash et al. .	
4,884,972	12/1989	Gasper	395/152 X
5,111,409	5/1992	Gasper et al. .	

FOREIGN PATENT DOCUMENTS

0065862	12/1982	European Pat. Off. .	
0584758A2	3/1994	European Pat. Off. .	
3-129572	6/1991	Japan .	
3-193074	8/1991	Japan .	
5184726	7/1993	Japan	273/434

2106789	4/1983	United Kingdom	273/434
2163929	3/1986	United Kingdom	273/85 G

OTHER PUBLICATIONS

Sam S. Rakover et al, Cognitive Psychology, vol. 21, pp. 423–464 (1989), To Catch a Thief with a Recognition Test: The Model and Some Empirical Results.

Patent Abstracts of Japan, vol. 018, No. 172 (P–1715) Mar. 23, 1994 & JP–A–05 334411 (Casio Computer Co., Ltd.) Dec. 17, 1993.

Primary Examiner—Jessica Harrison
Assistant Examiner—Michael O'Neill
Attorney, Agent, or Firm—Frishauf, Holtz, Goodman, Langer & Chick

[57] **ABSTRACT**

An electronic competing game apparatus utilizes article image data representative of human, animal, and housings to obtain a competing result of this game. In the electronic competing game apparatus, the respective portions of the article is designated by operating a designation unit, and a plurality of article images are displayed on a display unit. A competing result between the respective article images displayed on the display unit is judged by a judging unit based upon both point data corresponding to the partial images for constituting the respective article images, and point data corresponding to numeral data produced in an irregular sequence. When the designation unit is operated while the respective article images are attacking with each other, the irregularly set point data is added to the point data set to the respective article images. The competing results between the respective article images are determined based on the added point data.

22 Claims, 17 Drawing Sheets
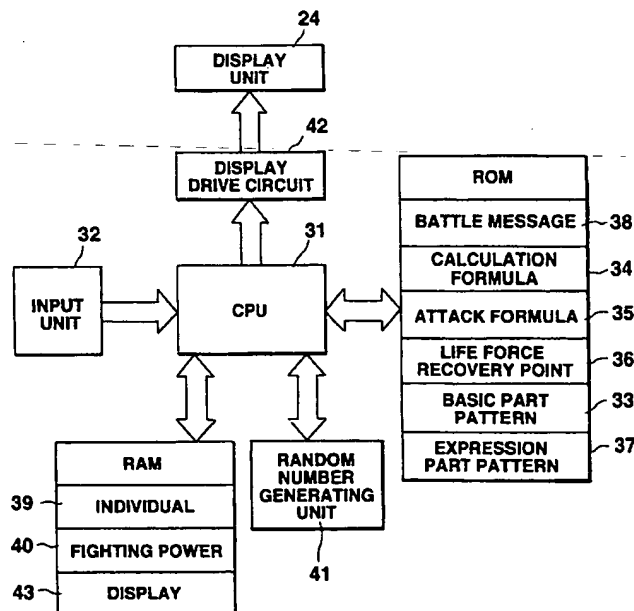
FIG.1
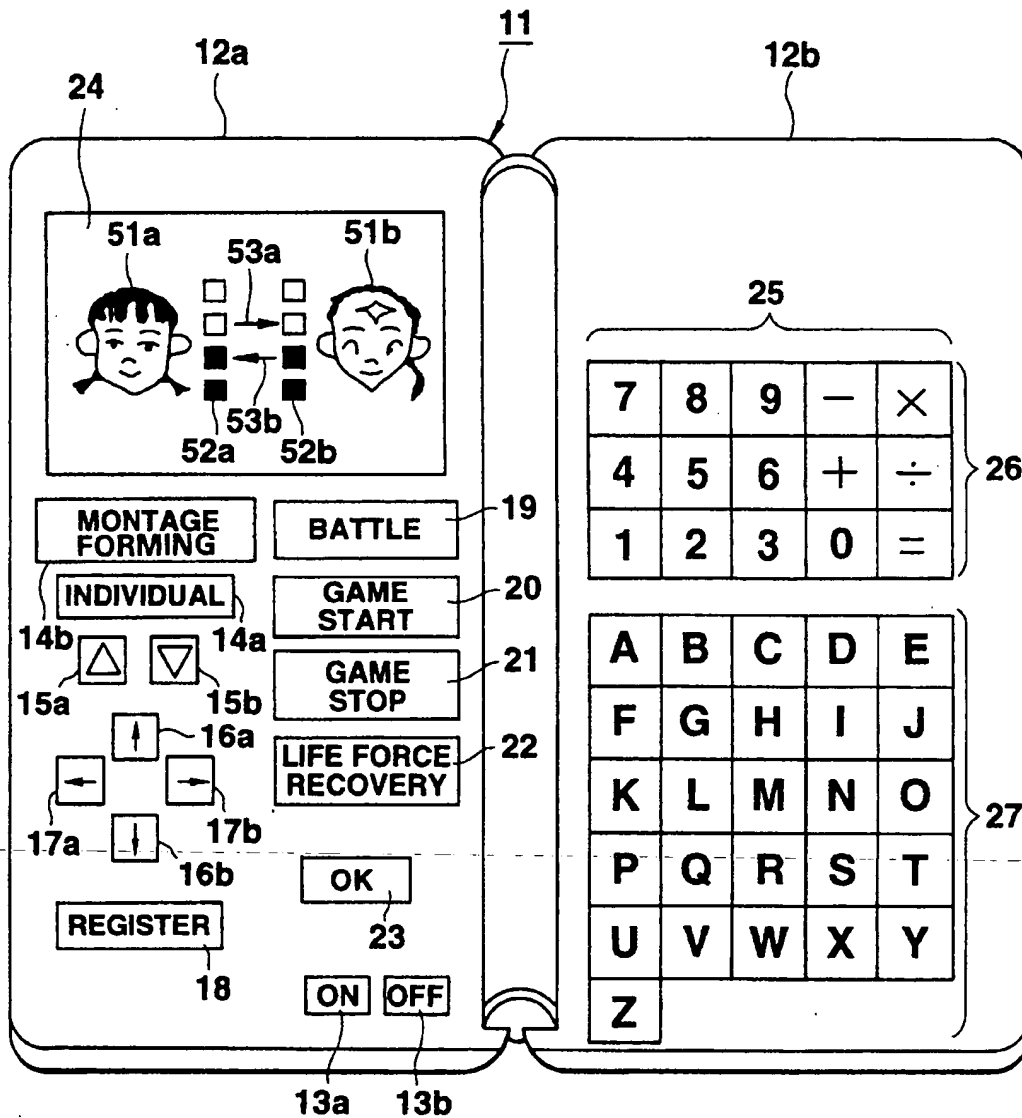
FIG.2
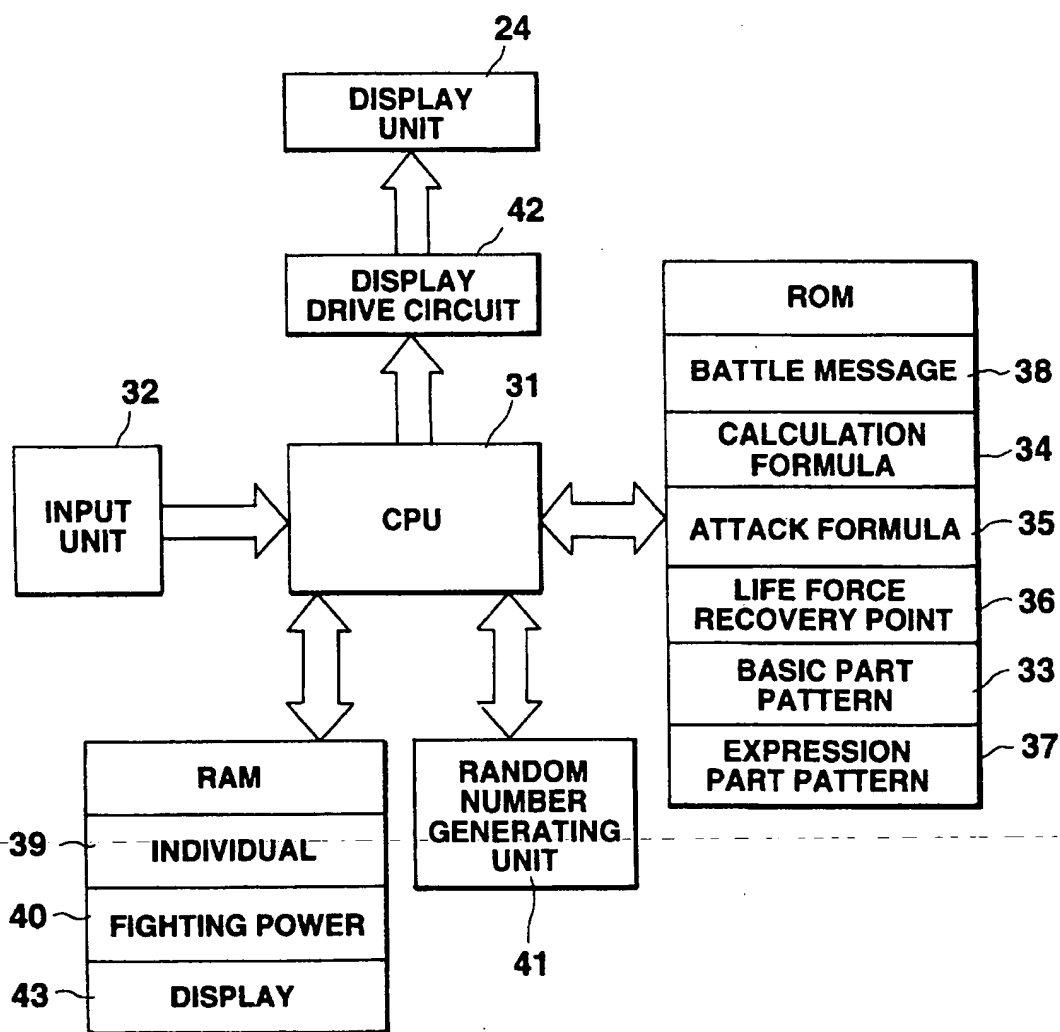
FIG.3

BASIC PART PATTERN ROM

33
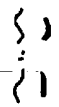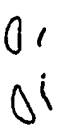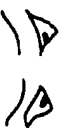
PARTS \ NO.		01	02	03	04	20
OUTLINE	1					
HAIR STYLE	2					
EYEBROWS EYES	3 · 4					
NOISE	5					
MOUTH	6					

FIG.4

LIFE FORCE RECOVERY POINT ROM

36

54a	ICON(1)	40000 POINT
54b	ICON(2)	30000 POINT
54c	ICON(3)	27000 POINT
54d	ICON(4)	23000 POINT
54e	ICON(5)	20000 POINT
54f	ICON(6)	17000 POINT
54g	ICON(7)	13000 POINT
54h	ICON(8)	10000 POINT
54i	ICON(9)	5000 POINT
54j	ICON(10)	0 POINT

FIG.5

DISPLAY RAM

DATA / ITEM	INDIVIDUAL DATA				MONTAGE DATA						
	NAME	ADDRESS	TELEPHONE	AGE	OUTLINE	HAIR STYLE	NOSE	MOUTH	WEAR	RIGHT/LEFT HANDS	RIGHT/LEFT FEET
1	A	○	○	40	101	202	504	601	703	NONE	NONE
2	B	○	○	30	105	206	503	604	705	NONE	NONE
3	C	○	○	15	110	211	515	618	720	820	920
4	D	○	○	20	105	220	510	611	710	815	914
50	Z	○	○	50	110	215	520	615	712	815	920

39a — 39b — 43

39

BASIC FIGHTING FORCE DATA (40a)

LIFE FORCE DATA	1	2	4	1	3	
ATTACKING FORCE DATA	0	0	0	0	0	
DEGENSIVE FORCE DATA	1	2	5	6	7	

FIGHTING FORCE COEFFICIENT (40b)

LIFE FORCE COEFFICIENT A (40b1) | ATTACKING FORCE COEFFICIENT B (40b2) | DEFENSIVE FORCE COEFFICIENT C (40b3)

DETERMINED FIGHTING POWER DATA (40c)

LIFE FORCE (ESP) (40c1) | ATTACKING FORCE (PK) (40c2) | DEFENSIVE FORCE (TP) (40c3)

DATA ON LIFE FORCE AFTER ATTACKING (40d)

DATA IN LIFE FORCE AFTER RECOVERY (40e)

40

FIG.6

(OVERALL PROCESS)

↓

| EXECUTE INPUT/REGISTER PROCESS OF INDIVIDUAL DATA (SEE FIG.7) | X1 |

↓

| EXECUTE MONTAGE FORMING/REGISTER PROCESS (SEE FIG.8 & FIG.9) | X2 |

↓

| EXECUTE REGISTER PROCESS OF FIGHTERS AND SETTING PROCESS OF FIGHTING POWER DATA (SEE FIG.10 & FIG.11) | X3 |

↓

| EXECUTE BATTLE GAME PROCESS (SEE FIG.12 & FIG.15) | X4 |

↓

X5

◇ IS GAME STOP KEY OPERATED ? ◇ —— NO

YES ↓

(END)

FIG.7

INDIVIDUAL DATA
INPUT/REGISTER PROCESS

Y1

IS
INDIVIDUAL KEY
TURNED ON
?

NO

YES

Y2

IS
INDIVIDUAL DATA
INPUTTED
?

NO

YES

Y3

STORE INDIVIDUAL
DATA INTO RAM

Y4

IS
REGISTER KEY
TURNED ON
?

NO

YES

END

FIG.8

(MONTAGE FORMING PROCESS)

S1 — IS MONTAGE FORMING KEY OPERATED ? — NO

YES

S2 — SET BASIC TYPE PART PATTERN NUMBER 01 SET CHANGING PART NUMBER 1

S3 — COMBINE PART PATTERN DATA (SEE FIG.9)

S4 — DISPLAY COMBINED PART PATTERN

S5 — IS △ ▽ KEY OPERATED ? — YES — S6 CHANGE TYPE PART PATTERN

NO

S7 — IS ↑ ↓ KEY OPERATED ? — YES — S8 CHANGE PART SORT

NO

S9 — IS ← → KEY OPERATED ? — YES — S10 CHANGE PART PATTERN

NO

S11 — IS REGISTER KEY OPERATED ? — NO

YES

S12 — REGISTER

(END)

FIG.9

DATA SYNTHESIZING PROCESS

A1
READ OUTLINE PART
PATTERN NUMBER

A2
STORE CORRESPONDING PART
PATTERN INTO DISPLAY RAM

A3
READ HAIR STYLE
PART PATTERN NUMBER

A4
STORE CORRESPONDING PART
PATTERN INTO DISPLAY RAM

A5
EXECUTE SIMILAR PROCESS
FOR OTHER PORTIONS, E.G.,
EYES, NOSE, MOUTH

A6
DISPLAY DESIRABLE
MONTAGE IMAGE

RETURN

FIG.10

```
   ┌─────────────────────────────┐
   │ FIGHTER REGISTER/FIGHTING   │
   │ POWER DATA SETTING PROCESS  │
   └─────────────────────────────┘
                  │
                  ▼            B1
            ◇ IS BATTLE KEY OPERATED ? ◇──NO──┐
                  │ YES                         │
                  ▼                             │
   ┌─────────────────────────────┐  B2         │
   │ DISPLAY LIST OF SELECTED    │             │
   │ FIGHTER'S NAME (FIG.16A)    │             │
   └─────────────────────────────┘             │
                  │                             │
                  ▼            B3               │
            ◇ IS FIGHTER SELECTED ? ◇──NO───────┘
                  │ YES
                  ▼
   ┌─────────────────────────────┐  B4
   │ DISPLAY PORTRAIT OF SELECTED│
   │ FIGHTER TOGETHER WITH HIS   │
   │ NAME (FIG.16B & FIG.16C)    │
   └─────────────────────────────┘
                  │
                  ▼            B5
            ◇ IS OK KEY OPERATED ? ◇──NO──┐
                  │ YES                    │
                  ▼                        
```

B6 READ MONTAGE DATA OF SELECTED FIGHTER FROM RAM, ALLOCATE LIFE FORCE, ATTACKING FORCE, DEFENSIVE FORCE TO EACH PART, AND STORE ALLOCATED ITEMS INTO FIGHTING FORCE DATA AREA OF RAM FOR SELECTED FIGHTER

B7 CALCULATE TOTAL VALUES FOR RECEPTOR LIFE FORCE DATA, ATTACKING FORCE DATA, DEFENSIVE FORCE DATA, AND SET FIRST DIGIT VALUE OF EACH TOTAL VALUE AS ATTACKING COEFFICIENT

(A)

FIG.11

(A)

B8
SUBSTITUTE ATTACKING
COEFFICIENT FOR CALCULATION
FORMULA READ FROM CALCULATION
ROM TO OBTAIN FIGHTING POWER
DATA (LIFE FORCE, ATTACKING
FORCE, DEFENSIVE FORCE)

B9
DISPLAY OBTAINED FIGHTING
POWER DATA TOGETHER WITH
PORTRAIT OF SELECTED FIGHTER
(FIG.16D)

B10
HAS 1
SECOND PASSED
?   NO

YES

B11
CALCULATE FIGHTING
POWER DATA BASED ON PRESET
MONTAGE DATA ABOUT
COMPUTER FIGHTER

B12
DISPLAY LIFE FORCE DATA VALUE
AND PORTRAIT OF
COMPUTER FIGHTER (FIG.16E)

B13
HAS 1
SECOND PASSED
?   NO

YES

B14
GENERATE RANDOM NUMBER
AND DETERMINE WHICH
FIGHTER ATTACKS FIRST

B15
DISPLAY ARROWS CORRESPONDING
TO DETERMINED FIRST ATTACKER
AND SECOND ATTACKER, BAR
GRAPHS AND PORTRAITS
CORRESPONDING TO RESPECTIVE
FIGHTING POWER DATA (FIG.16F)

RETURN

FIG.12

BATTLE GAME PROCESS

C1
IS GAME START KEY OPERATED ? — NO

F → YES

C2
DISPLAY FIGURE OF FIRST ATTACKER UNDER ATTACKING (FIG.16G)

C3
GENERATE RANDOM NUMBER, AND SUBSTITUTE GENERATED RANDOM NUMBER (0~9) FROM ATTACKING FORMULA READ FROM ATTACKING FORMULA ROM TO OBTAIN ATTACKING POINT VALUE OF FIRST ATTACKER

C4
CALCULATE NEW LIFE FORCE DATA VALUE OF SECOND ATTACKER BASED ON CALCULATED ATTACKING POINT VALUE AND LIFE FORCE DATA VALUE BEFORE ATTACKING

C5
DISPLAY BAR GRAPH, PORTAIT, ARROW OF SECOND ATTACKER CORRESPONDING TO CALCULATED LIFE FORCE DATA (FIG.16H)

C6
HAS PRESELECTED SECOND PASSED ? — NO

YES

C7
DISPLAY FIGURES OF SECOND ATTACKER UNDER ATTACKING (FIG.16I)

B

FIG.13

(B)

C8

GENERATE RANDOM NUMBER, AND SUBSTITUTE GENERATED RANDOM NUMBER FOR CALCULATION FORMULA READ FROM ATTACKING FORMULA ROM TO OBTAIN ATTACKING POINT VALUE OF SECOND ATTACKER TO OBTAIN ATTACKING POINT VALUE OF SECOND ATTACKER

C9

CALCULATE NEW LIFE FORCE DATA VALUE OF FIRST ATTACKER BASED ON CALCULATED ATTACKING POINT VALUE AND LIFE FORCE BEFORE ATTACKING

C10

DISPLAY BAR GRAPH, PORTRAIT, AND ARROW OF FIRST ATTACKER, CORRESPONDING TO CALCULATED LIFE FORCE DATA VALUE

C11

IS LIFE FORCE RECOVERY KEY OPERATED ?   NO

YES

C12

EXECUTE LIFE FORCE RECOVERY PROCESS

C13

NO   (F)   IS LIFE FORCE "0" ?   (SELECTED FIGHTER "0")   (E)

(COMPUTER "0")   (D)

C14

DISPLAY PORTRAIT CORRESPONDING TO LIFE FORCE DATA VALUE, CORRESPONDING BAR GRAPH, AND PORTRAIT WHEN COMPUTER FIGHTER LOST GAME (FIG.17A)

C15

ERASE COMPUTER FIGHTER (FIG.17B)

(C)

FIG.14

C16
IS OK KEY OPERATED ? — NO → D

YES

C17
DISPLAY POTRAIT OF ANOTHER COMPUTER FIGHTER TOGETHER WITH NAME (FIG.17C)

C19
DISPLAY PORTRAIT CORRSPONDING TO LIFE FORCE DATA VALUE "0", CORRESPONDING BAR GRAPH, PORTRAIT WHEN SELECTED FIGHTER LOST GAME (FIG.17E)

C18
EXECUTE PROCESS OF SIMILAR ATTACKING CONDITION

C20
ERASE IMAGE OF SELECTED FIGHTER (FIG.17F)

(RETURN)

FIG.15

```
        ( LIFE FORCE RECOVERY PROCESS )
                      │
                      ▼◄──────────────────┐
                   ╱ IS ╲           D1     │
              ╱ LIFE FORCE ╲               │
             ◄ RECOVERY KEY OPERATED ►─ NO ┤
              ╲     ?    ╱                 │
                 ╲    ╱                    │
                 YES                       │
                  │◄──────────────────┐    │
                  ▼              D2    │    │
        ┌──────────────────────┐      │    │
        │ DISPLAY SUB-MENU (FIG.16K) │ │    │
        └──────────────────────┘      │    │
                  │            D3      │    │
                  ▼                    │    │
               ╱ IS ╲                  │    │
          ◄ ITEM SELECTED ►──── NO ────┘    │
               ╲  ?  ╱                       │
                 YES                         │
                  │              D4          │
                  ▼                          │
        ┌──────────────────────┐            │
        │ GENERATED RANDOM NUMBER │          │
        └──────────────────────┘            │
                  │◄───────────────────┐     │
                  ▼                     │     │
   ┌──────────────────────────────┐    │     │
   │ ROTARY DISPLAY ICON FROM ICONS IN │ │     │
   │ ACCORDANCE TO GENERATED RANDOM    │ │     │
   │ NUMBER IN PRESELECTED SEQUENCE(FIG.16L)│ │
   └──────────────────────────────┘  D5 │     │
                  │         D6           │     │
                  ▼                      │     │
               ╱ IS ╲                    │     │
          ◄ OK KEY OPERATED ►──── NO ─────┘     │
               ╲  ?  ╱                          │
                 YES                            │
                  │              D7              │
                  ▼                             │
        ┌──────────────────────┐               │
        │ STOP ICON (FIG.16M)   │         D8    │
        └──────────────────────┘               │
                  │                             │
                  ▼                             │
   ┌──────────────────────────────┐            │
   │ SUBSTITUTE POINT VALUE OF STOPPED ICON │   │
   │ FOR LIFE FORCE RECOVERY FORMULA READ   │   │
   │ FROM LIFE FORCE RECOVERY FORMULA ROM   │   │
   │ TO OBTAIN NEW LIFE FORCE DATA VALUE    │   │
   └──────────────────────────────┘            │
                  │
                  ▼
             ( RETURN )
```

FIG.16A

FIGHTER
SELECTED ?
X
Y
A
24

FIG.16B

FIGHTER
SELECTED ?
A
24

FIG.16C

FIGHTER
SELECTED ?
A
24
51a

FIG.16D

ESP 50000
PK   4000
TP   3000
24
51a

FIG.16E

LEVEL-1
ESP 40000
PK   ????
TP   ????
24
51b

FIG.16F

53a
24
51a   52a   52b   51b

FIG.16G

DISPLAY ATTACKING
RESULT GIVEN
FROM DEFENDER
24

FIG.16H

53b
24
51a   52a   52b   51b

FIG.16I

DISPLAY ATTACKING
RESULT GIVEN
FROM ATTACKER
24

FIG.16J

53a
24
51a   52a   52b   51b

FIG.16K

◆◆   SUB-MENU   ◆◆
▶ ESP ITEM
CONTINUE FIGHTING
24

FIG.16L

53a
ROTATION
24
54   52a   52b   51b

FIG.16M

53a
27000
24
54c   52a 53a   52b   51b

FIG.16N

24
51a   52a   52b   51b

< CASE WHERE FIRST
COMPUTER FIGHTER
LOST GAME >

FIG.17A

51a   52a   52b   51b

FIG.17B

51a   52a

FIG.17C

LEVEL-2
ESP 50000
PK   ????
TP   ????

51c

< CASE WHERE SECOND
COMPUTER FIGHTER
LOST GAME >

FIG.17D

51a   52a

< CASE WHERE
SELECTED FIGHTER
LOST GAME >

FIG.17E

51a   52a   52b   51b

FIG.17F

52b   51b

1

ELECTRONIC BATTLE GAME PLAYING APPARATUS WITH FACIAL MONTAGE GENERATION

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to an electronic game playing method and an electronic game playing apparatus for judging a competing result with employment of article image data such as human, animals and housings. More specifically, the present invention is directed to a method and an apparatus capable of electronically playing a battle game, while displaying a plurality of fighters whose life powers are selected by a user on a display screen.

2. Description of the Prior Art

Various sorts of game apparatuses with utilizing electronic appliances have been developed and are being marketed. For instance, the battle type television game apparatuses equipped with TV screens are widely spread. In the typical TV game apparatus, the game program data previously stored in the specific cassette tape magazine mounted on this TV game apparatus are read in response to the key operations, and then a user moves the character images displayed on the TV screen in accordance with this game program data, finally, the microcomputer judges the competing result between these character images.

In this conventional TV game apparatus, the data used to judge the competing result are such data previously set in this cassette magazine as a portion of the game program. As a result, the conclusions of this competing result are predetermined, so that amusing TV game cannot be expected.

To solve the above-explained drawback, U.S. patent application Ser. No. 148,975 entitled to T. MASE et al., "IMAGE DATA PROCESSOR" assigned to the same assignee as that of the present U.S. patent application, has proposed such an electronic game apparatus for playing the battle game with use of the article image data, e.g., a plurality of human, animals, or housings. In this electronic game apparatus, a plurality of article image data are produced by combining the portional image data with each other with regard to the respective portions, and a decision is made of the competing result between the respective article images based upon a comparison result in the magnitudes of the point data set in correspondence with the respective portion image data for constituting these plural article images.

However, there is a recent trend that users want to enjoy the battle game with great flexibility. That is, since the competing result between the respective article images is determined based upon the comparison result in the magnitudes of the point data set in correspondence with the respective partial image data which constitute the respective article image data, the same point data are always allocated to the specific article image constructed from the same partial image data. As a consequence, while any users continuously utilize the same electronic game apparatus, they can gradually become aware of the competing results with respect to the same article image, so that they may not deeply enjoy this battle game with their great fun.

SUMMARY OF THE INVENTION

The present invention has been made in an attempt to solve the above-described problems, and therefore, has an object to provide a novel electronic game playing method

2

and an apparatus capable of playing a game with great fun.

Another object of the present invention is to provide an electronic game playing method and an apparatus capable of obtaining an unexpected competing result.

A further object of the present invention is to provide an electronic game playing method and an apparatus capable of reflecting sufficient intention of a game player in the game.

To achieve these objects, a method for electrically playing a competing game on a display unit, according to one aspect of the present invention, is featured by comprising the steps of:

producing a plurality of numeral data having different values in an irregular sequence;

setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles;

judging a competing result between a plurality of article images represented by said plurality of article image data based upon both point data set in correspondence with the portion image data for constituting said article image data, and point data corresponding to said numeral data; and

displaying said competing result between said article images on the display unit.

Furthermore, an electronic game apparatus for electronically playing a competing game, according to another aspect of the present invention, is featured by comprising:

numeral data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles;

judging means for judging a competing result between a plurality of article images represented by said plurality of article image data based on both point data set in correspondence with the portion image data, and point data corresponding to said numeral data; and

display means for displaying said competing result between said article images.

The above-described display means may be selected from a liquid crystal display device, a printer, and the like which can visually represent the competing result.

According to the present invention, a plurality of article image data made of combinations of the partial image data corresponding to the respective portions of the article may be set by the setting means. Subsequently, the competing result between the plural article images set by this setting means may be judged by the judging means based on both the point data set in correspondence with the partial image data for constituting this article image, and the point data corresponding to the numeral data generated from the numeral value data generating means in such an irregular sequence. This judgement result is displayed on the display means. In this case, the judgement result may be made by taking account of the point data corresponding to the numeral data generated by the numeral value data generating means in the irregular sequence in addition to the point data set in correspondence with the partial image data for constituting the respective article image data. As a result, the unexpected competing results between the plural article images could be achieved in the above-defined electronic game playing apparatus/method according to the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention, reference is made of the following detailed description to be

read in conjunction with the accompanying drawings, in which:

FIG. 1 is an outer view of a montage battle game machine according to an embodiment of the present invention;

FIG. 2 is a schematic block diagram for representing an overall arrangement of an electronic circuit employed in the montage battle game machine shown in FIG. 1;

FIG. 3 schematically illustrates storage conditions of basic part patterns stored in a basic part pattern ROM;

FIG. 4 schematically shows storage conditions of life force recovery points stored in a life force recovery point ROM;

FIG. 5 schematically shows storage conditions of various sorts of data stored in a RAM;

FIG. 6 is a flow chart for explaining an overall process operation of the montage battle game machine shown in FIG. 1;

FIG. 7 is a flow chart for explaining an input process operation of individual data;

FIG. 8 is a flow chart for explaining a montage forming process operation;

FIG. 9 is a flow chart for representing a data synthesizing process operation executed in conjunction with the above-explained montage forming process operation;

FIG. 10 is a flow chart for explaining a front half process operation about registering process of battle fighters/setting process of fighting power;

FIG. 11 is a flow chart for explaining a rear half process operation about registering process of battle fighters/setting process of fighting power;

FIG. 12 is a flow chart for indicating a front stage of the battle Game process operation;

FIG. 13 is a flow chart for indicating a middle stage of the battle Game process operation;

FIG. 14 is a flow chart for indicating a rear stage of the battle game process operation;

FIG. 15 is a flow chart for representing a life force recovery process operation executed during the battle Game processing operation;

FIGS. 16A to 16N schematically represent display conditions appearing during the battle fighter registering process, the battle game process, and the life force recovery process; and

FIGS. 17A to 17F schematically indicate display conditions appearing in the respective win/lose determining stages during the battle Game process.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to drawings, a montage battle Game apparatus according to a presently preferred embodiment of the present invention will be described.

OVERVIEW OF MONTAGE BATTLE GAME APPARATUS

FIG. 1 schematically shows an outer view of a notebook type montage battle game apparatus 11 to which the present invention has been applied.

The montage battle game apparatus 11, as illustrated and viewed in FIG. 1, owns a notebook type housing operable in the right/left directions. That is, this notebook type housing is constructed of a left operation plane 12a and a right

operation plane 12b. On the left operation plane 12a, there are provided: an ON key 13a and OFF key 13b used to turn ON/OFF a power source (not shown in detail); an individual key 14a operated when an individual data input mode is set; and a montage forming key 14b manipulated when a montage forming mode is set. This left operation plane 12a of the montage battle game apparatus further contains basic montage selection switches 15a and 15b operated when the basic montage made of combinations of various basic part patterns previously stored is selected to form a desired montage; keys 16a and 16b manipulated as part designation keys when a part to be changed is designated to form a desirable montage, and also as fighter selection keys when a fighting fighter is selected during the battle game; and pattern selection keys 17a and 17b operated when a part pattern about the part to be changed, which has been designated as the part to be changed; during the formation of the above-explained desired montage. This left operation plane 12a further includes a register key 18 operated when either the entered individual data, or the formed montage pattern is reserved; a battle key 19 manipulated when a battle fighter register is set; a game start key 20 operated when a battle game is started; a game stop key 21 operated when the battle game is stopped, or ended; a life force recovery key 22 operated when a life force recovery mode is set by which life force of a selected fighter is recovered during a battle game; and an OK key 23 manipulated as a selection/execution key in the respective battle fighter register mode, battle game mode, and life force recovery mode.

Also, a liquid crystal dot matrix display unit 24 is provided on the left operation plane 12a of the main body of this game apparatus 11.

As illustrated in FIG. 1, the liquid crystal dot matrix display unit 24 is mainly arranged by one montage display area used to display a selected fighter 51a which has been registered by a user in the battle fighter register mode in response to the key operation of the above-described battle key 19, and the other montage display area used to display a computer fighter 51b which has been previously registered. Furthermore, there are provided inbetween these display areas, a life force display bar 52a for the selected fighter, used to display a remaining amount of life force of each fighter; another life force display bar 52b for life force of the computer fighter, and attacking arrows 53a, 53b indicative of attacking directions of the respective fighters.

On the other hand, the right operation plane 12b of the game apparatus body 11 contains numeral entry keys ("0" to "9") 25 operated when a numeral value is entered; operator keys 2b; and alphabetical keys 27 used to enter individual data and various sorts of data.

ELECTRONIC CIRCUIT OF MONTAGE BATTLE GAME APPARATUS

FIG. 2 is a schematic block diagram for showing an overall circuit of the above-explained montage battle game apparatus according to the first embodiment of the present invention.

This montage battle game apparatus includes a CPU (central processing unit) 31. The CPU 31 performs operation controls of various circuit units in accordance with the previously stored program in response to various key operation signals supplied from the input unit 32 provided on the left operation plane 12a and the right operation plane 12b. To this CPU 31, there are connected in addition to the above-explained input unit 32, a basic part pattern ROM 33;

a calculation formula ROM **34**; an attack formula ROM **35**; a life force recovery point ROM **36**; an expression part pattern ROM **37**; a battle message ROM **38**; individual data RAM **39**, fighting power data RAM **40**; a random number generating unit **41**; and via a display drive unit **42** to the above-described liquid crystal dot matrix display unit **24**.

VARIOUS DATA/PART PATTERN STORAGE CONDITIONS

FIG. **3** schematically represents storage conditions of basic part patterns previously stored in the basic part pattern ROM **33**.

20 sorts of part patterns have been stored into preselected storage areas numbered by No. 01 to NO. 20 in this basic part pattern ROM **33**. That is, 9 sorts of various parts components such as an outline, a hair style, eyebrows, eyes, a nose and a mouth are prepared for constituting various images of human, animals and a creature from outer space with respect to 20 sorts of parts patterns.

In this case, 20 sorts of predetermined basic montage images may be produced by combining the part patterns corresponding to the respective parts (outline, hair style, eyebrows, eyes, nose, mouth) in the basic part patterns No. 1 to No. 20 stored in the basic part pattern ROM **33**.

FIG. **4** schematically indicates storage conditions of life force recovery points previously stored in the life force recovery point ROM **36**.

Icons **54a** to **54j** corresponding to 10 sorts of image pattern data, which are selected by a user in the life force recovery mode, and also life force recovery points having present different values corresponding to these different pictorial icons **54a** to **54j**, are previously stored in this life force recovery point ROM **36**.

FIG. **5** schematically shows storage conditions of various data temporarily stored in the above-described RAMs.

That is, the RAMs include the individual data RAM **39**, the fighting power data RAM **40**, and the display RAM **43**. This individual data RAM **39** is arranged by an individual data register **39a** and a montage data register **39b**.

The individual data register **39a** is designated to store therein the individual data about 50 individuals which are arbitrarily inputted during the individual data input data. Each of these individual data is constructed of a name, an address, a telephone number, and an age. The montage data register **39b** is designated to store therein montage data corresponding to the respective individual data in the montage forming mode by combining part pattern numbers which correspond to a plurality of part patterns for constituting various montages (pictorial images).

In this case, the part numbers representative of the various parts (portions) are stored as the upper digit, whereas the part pattern numbers are stored as the lower digit. For instance, in case of individual data "A", since the portion of the outline is "1", and the selected part pattern number is "01", the montage data about this outline becomes "101". Furthermore, since the portion of the hair style is "2", and the selected part pattern number is "02", the montage data about this hair style becomes "202". Thus, the part numbers may be stored as the montage data for the respective parts in such a manner.

On the other hand, the fighting power data RAM **40** is arranged by a basic fighting power data register **40a**, a fighting power determining coefficient register **40b**, a determined fighting force data register **40c**, a data register **40d** for

life force after attacking, and a data register **40e** for life force after recovery.

The basic fighting force data register **40a** stores therein fighting force data which constitutes a base of a selected fighter, and has been registered by a user in the battle fighter register mode. The fighting power data is classified into life force data, attacking force data, and defensive force data, which will then be stored into the basic attacking force data register **40a**. In this case, two digits of the lower digit portion of the corresponding montage data are allocated to the life force data and the attacking force data, whereas the first digit of the upper digit portion thereof is allocated to the defensive force data.

For example, when an "A" is registered as the selected fighter in the individual data RAM **39**, since the montage data corresponding to the portion of the outline is "101", the above-described fighting power data becomes "1", "0", "1" in the order of the defensive force data, the attacking force data, and the life force data. Also, since the montage data corresponding to the portion of the hair style is "202", the above-mentioned fighting power data becomes "2", "0", "2" in the order of the defensive force data, the attacking force data, and the life force data. These numeral values are stored as the respective data.

Then, the fighting power determining coefficient register **40b** is subdivided into a life force coefficient register **40b1**, an attacking force coefficient register **40b2**, and a defensive force coefficient register **40b3**. Into the life force coefficient register **40b1**, the first digit value of the total value about the life force data allocated to the basic fighting power data register **40a** is stored as the life force determining coefficient "A" of the selected fighter. Into the attacking force coefficient register **40b2**, the first digit value of the total value about the attacking force data is stored as the attacking force coefficient "B" of the selected fighter. Furthermore, into the defensive force coefficient register **40b3**, the first digit value of the total value of the defensive force data is stored as the defensive force determining coefficient "C" of the selected fighter.

When the total value of the life force data stored in the basic fighting power data register **40a** is, for instance, "123", the above-explained life force determining coefficient "A" is "3" corresponding to the first digit value of this total value "123". When the total value of the attacking force data stored in the basic fighting power data register **402** is, for example, "26", the attacking force determining coefficient "B" is "6" corresponding to the first digit value of this total value "26". Also, when the total value of the attacking force data stored in the basic fighting power data register **40a** is, e.g., "135", the attacking force determining coefficient "C" is "5" corresponding to this total value of "135".

Also, the determined fighting power data register **40c** is subdivided into a life force data register **40c1**, an attacking force data register **40c2**, and a defensive force data register **40c3**. Into the respective registers **40c1**, **40c2**, and **40c3**, life force data (ESP) of the selected fighter, attacking force data (PK) thereof, and defensive force data (TP) thereof are stored which have been calculated based on a calculation formula for determining the fighting power data previously stored into the calculation formula ROM **34**. In this case, the calculation formula to determine the life force is given by the below-mentioned formula (1):

$$\text{Life force data } (ESP) = \text{Life force determining coefficient "}A\text{"} \times 10,000 \tag{1}.$$

7

The calculation formula to determine the attacking force is given by the following formula (2):

$$\text{Attacking force data } (PK) = \text{Attacking force determining coefficient } "B" \times 1,000 \quad (2).$$

Further, the calculation formula to determine the defensive force is given by the following formula (3):

$$\text{Defensive force data } (TP) = \text{Defensive force determining coefficient } "C" \times 1,000 \quad (3).$$

As a consequence, when the life force determining coefficient "A" is "3", which is stored in the fighting power determining coefficient register 40b in correspondence with the selected fighter "A", the life force data (ESP) thereof becomes (30,000) calculated based on the above-explained formula (1). When the attacking force determining coefficient "B" is "6", the attacking force data (PK) thereof becomes (6,000) calculated based upon the above-described formula (2). Furthermore, when the defensive force determining coefficient "C" is "5", the defensive force data (TP) becomes (5,000) calculated based on the above-mentioned formula 3. These numeral values are stored into the respective registers 40c1 to 40c3 for constituting the determined fighting force data register 40c.

The data register 40d for life force after attacking stores therein the remaining amount of the life force data after the attacking from the confronted fighter. In this case, the remaining amount of the life force after the respective fighters have attacked, is calculated based on a calculation formula (4) about the life force data after attacking, which has been previously stored in the attacking formula ROM 35:

$$\text{Life force data after attacking} = \text{Life force before attacking} - \text{Attacking point for confronted fighter } (BP) \quad (4).$$

It should be noted that the attacking point for attacking fighter (BP) may be calculated based on the following formulas (5) and (6). The first formula (5) is a basic value calculating formula which has been previously stored in the attacking formula ROM 35:

$$\text{Basic value} = \text{Attacking force data for attacking fighter} \times (1 - \text{Defensive force for confronted fighter}/20,000) \times 4 \quad (5).$$

Also, the second formula (5) is an attacking point calculating formula:

$$\text{Attacking point } (BP) = \text{Basic value} \times \text{Random numeral value} \quad (6).$$

The data register 40e for life force after recovery stores therein the life force data recovered in accordance with the life force recovery point (refer to FIG. 4) selected in the life force recovery mode. This life force data after recovery is updated as life force data (ESP) of the determined fighting force data register 40c.

Here, the above-described life force after recovery is calculated based on a life force recovery formula (7) previously stored in the calculation formula ROM 24:

$$\text{Life force after recovery} = \text{Original life force} + \text{Life force recovery point} \quad (7)$$

That is, both of the newly entered individual data and the montage data thereof are stored in the individual data RAM 39. The fighting power of the selected fighter is stored as the life force data (ESP), the attacking force data (PK), and the defensive force data (TP) in the fighting power data RAM 40.

8

It should be understood that a plurality of montage data about the fighters on the side of the computer, who are confronted with the fighters selected by the user, have been previously stored in a ROM (not shown), and then the fighting power data, the respective determining coefficients for life force, attacking force, and defensive force, and the various data about life force data (ESP), attacking force data (PK), defensive force data (TP), and the life force after attacking data, which belong to the computer fighters successively selected as the confronted fighters, can be obtained in a similar manner to those of the above-described selected fighters.

On the other hand, display part patterns at the respective battle win/lose determining stage have been stored in the expression part pattern ROM 37. That is, face expression changing part patterns corresponding to the eyes and mouth basic part patterns in the basic part pattern ROM 33 have been previously stored.

Also, win/lose messages varied in response to the battle results have been stored in the battle message ROM 38 in addition to the message data required during the battle game. For instance, a message "You did it!" has been previously stored as the message displayed in combination with the face montage for the winner, and another message "Sorry, try again!" has been previously stored as the message displayed in combination with the face montage for the loser.

On the other hand, the respective part patterns read out from the basic part pattern ROM 33 are synthesized in correspondence with the part pattern numbers for the respective parts which constitute the montages of the selected fighters and the computer fighters stored in the corresponding individual data RAM 39 and computer fighter ROM (not shown), and thereafter the combined part patterns are stored in the display RAM 43. A portrait image and the like are supplied via the display drive circuit 42 to the liquid crystal dot matrix display unit 42 as to be displayed thereon adjacent to the above-described part patterns (see FIG. 1).

OVERALL PROCESS OPERATION OF MONTAGE BATTLE GAME APPARATUS

Overall process operation of the above-described montage battle game apparatus with these arrangements will now be described with reference to a flow chart shown in FIG. 6.

Before executing the montage battle game, an input/register process operation (see FIG. 7) of individual data is carried out at a first step X1. At this step X1, arbitrary individual data is inputted as his name, address, telephone number, and age. At the next step X2, a montage forming process operation (see FIG. 8 and FIG. 9) is carried out, in which a portrait image of the respective individual data is formed and montage data corresponding to this individual data is registered.

Thereafter, at a step X3, a battle fighter register process operation (see FIG. 10 and FIG. 11) is performed, and furthermore another process operation is executed so as to set fighting power data about a selected fighter and a computer fighter. After the above-described various process operations have been executed, a battle game process operation (see FIG. 12 to FIG. 15) is carried out at a step X4. At this step X4, the battle game for the above-explained selected fighter battled with the computer fighter.

This battle game is performed until the game stop key 21 is manipulated by repeatedly executing the above-explained battle game process operation (steps X5 to X4).

INDIVIDUAL DATA INPUT/REGISTER PROCESS OPERATION

In FIG. 7, there is shown a flow chart for explaining the contents of the above-described individual data input/register process operation as defined at the step X1 of FIG. 6.

At a first step Y1 of this flow chart, when a judgement result is made that the individual key 14a in the input unit 32 is operated, the CPU 31 sets the operation mode to the individual data input mode. In this individual data input mode, when it is so judged at a step Y2 that such individual data as a name, an address, a telephone number, and an age are entered in accordance with the operations of the numeral entry key 25 and the alphabet entry key 27, these entered individual data are sequentially stored into the individual data register 39a in the individual data RAM 39 at a step Y3.

At the subsequent step Y4, when the register key 18 is manipulated, the individual data input mode is released, so that registers of the respective individual data stored in the individual data register 39a are confirmed.

MONTAGE FORMING/REGISTER PROCESS OPERATION

FIG. 8 is a flow chart for representing the contents of the above-explained montage forming/register process operation as defined in the step X2 of FIG. 6.

At a first step S1 of this flow chart, when the montage forming key 14b employed in the input unit 32 is operated, the operation mode of the CPU 31 is brought into the montage forming mode.

When the montage forming mode is in effective, the part pattern number "01" corresponding to the respective part patterns for constituting the first basic type montage image is set to the basic part pattern ROM 33 (see FIG. 3) at a step S2. Also, the first part number (namely, part number of outline in this case) is set as the part used to change the pattern.

As a consequence, the respective part patterns corresponding to the first basic type montage image are read out from the basic part pattern ROM 33. Thereafter, these read part patterns are transferred to the display RAM 43 and are combined with each other at the next step S3. As a result, the combined part patterns are displayed as the first basic type montage image on the liquid crystal dot matrix display unit 24 at a step S4.

Referring now to a flow chart of FIG. 9, a detailed process operation of the above-described data combining (synthesizing) process operation as defined at the previous step S3 of FIG. 8 will be explained.

In this data synthesizing process, when the part pattern numbers of the respective parts are designated to the basic part pattern ROM 33 in order to constitute the first basic type montage image, the part pattern data corresponding to the designated part pattern numbers are read out from the basic part pattern ROM 33, and then are transferred to the display RAM 43 in which the contents of these part pattern data are synthesized with each other (steps A1 to A5).

In this case, the respective part patterns such as the outline, eyes, and nose which constitute the first basic type montage image are sequentially synthesized in the display RAM 43. As a result, with respect to this initial setting operation, the first basic type montage image constructed by the respective parts and the respective part patterns corresponding to the part pattern number "01", is displayed on the liquid crystal dot matrix display unit 24 (step A6).

Under this condition, when the basic montage selecting key 15b employed in the input unit 32 is operated, all of the respective part pattern numbers for designating the basic type montage image to the basic part pattern ROM 33 are changed from "01" to "02" (steps S5 and S6).

As a consequence, in accordance with the changed part pattern number "02", after the respective part pattern data stored in the basic part pattern ROM 33 have been read out (in this case, all numbers are "02"), the transfer/synthesizing process operation of the respective part patterns with respect to the display RAM 43 is carried out. Accordingly, the second basic type montage image is displayed on the liquid crystal dot matrix display unit 24 at the steps S3 and S4. This second basic type montage image is constructed by synthesizing the part patterns corresponding to the part pattern number "02" which involves all of the part patterns from the outline part pattern to the mouth part pattern.

When the basic montage selection keys 15a and 15b employed in the input unit 32 are sequentially operated, the process operation defined at the steps S3 to S6 is repeatedly carried out, so that 20 sorts of basic type montage images previously stored as the part pattern numbers of "01" to "20" in the basic part pattern ROM 33 are sequentially changed and sequentially synthesized with each other, and finally are displayed.

As a consequence, the user can previously select either such a desirable montage image itself, or one basic type montage image similar to this desirable montage image to be displayed on the liquid crystal dot matrix display unit 24.

Under this display condition, when the part designation keys 16a and 16b employed in the input unit 32 are operated, the parts whose patterns should be changed, will be changed into other parts. For instance, when a specific part (for example, outline part) among the basic type montage images selectively and previously displayed on the liquid crystal dot matrix display unit 24 by executing the process operation defined at the steps S3 to S6, is desired to be changed into another different part (e.g., hair part), this specific part is changed by operating the part designation keys 16a and 16b (steps S7 and S8).

Thereafter, when the pattern selection keys 17a and 17b are operated, the part pattern is changed among the corresponding parts of the basic part pattern ROM 33, which are designated as the parts to be changed. This changed part pattern is read out from the basic part pattern ROM 33 and then transferred to the display RAM 43. Thus, the presently displayed part pattern on the liquid crystal display unit 24 may be substituted by the transferred part pattern, and then this transferred part pattern is displayed thereon (steps S9, S10, to S3, S4). As a consequence, the user may change the desired part into another different part among the basic type montage images selectively, previously displayed on the liquid crystal display unit 24, and then may display this different part on this display unit 24.

Furthermore, when this different part is changed into the other part pattern, the part designation keys 16a and 16b are operated to change the part to be change. Thereafter, the pattern selection keys 17a and 17b are operated to change the part pattern number, whereby the part among the presently displayed basic type montage images is changed into other part pattern, and then a desired montage image can be finally formed. At the steps S5 through S10, for instance, as the respective part patterns of the montage image corresponding to the individual data "A", the outline part pattern is selected as "01"; the hair style part pattern is selected as "02"; . . . , the nose part pattern is selected as "04"; the

mouth part pattern is selected as "01"; and also the wear part pattern is selected as "03". After the montage image desired by the user could be obtained, when it is judged at a step S11 that the register key 18 of the input unit 32 is operated, the part pattern numbers corresponding to the respective part patterns for constituting the synthesized montage image in the display RAM 43 are registered into the montage data register 39b within the individual data RAM 39 as the montage data corresponding to, for instance, the first individual data "A" at the step S12.

REGISTER PROCESS OPERATION OF BATTLE FIGHTER

FIG. 10 and FIG. 11 are flow charts for explaining the detailed contents of the battle fighter register process operation as defined at the above-described step X3 shown in FIG. 6.

At a first step B1 of the flow chart shown in FIG. 10, when a judgement result is made that the battle key 19 employed in the input unit 32 is operated during the battle fighter register process operation, the operation mode of the CPU 31 is set to the battle fighter register mode. Thereafter, the names of the respective individual data stored in the individual data register 39a of the individual data RAM 39 are transferred to the display RAM 43 in a predetermined format at the next step B2. Then, as shown in FIG. 16A, these individual names are displayed as a list of fighters to be selected, on the liquid crystal dot matrix display unit 24 in combination with such a character data "IS FIGHTER SELECTED" read out from the battle message ROM 38.

Under such a display condition, the fighter selection keys 16a and 16b are operated at a step B3 to select "A" as the selected fighter 51a, as illustrated in FIG. 16B. Accordingly, the montage data corresponding to this individual name "A" is read out from the montage data register 39b of the individual data RAM 39 at a step B4, so that a portrait of this selected fighter 51a is displayed in combination with his name "A" (see FIG. 16C).

In case of this portrait, since the respective part pattern numbers stored in the montage data register 39b in correspondence with the selected fighter whose name is "A", are given such that the outline is "01", the hair style is "02", . . . , the nose is "04" the mouth is "01", . . . , the corresponding part patterns are read out from the basic part pattern ROM 33 in accordance with these part pattern numbers. Then, the read part patterns are synthesized with each other in the display RAM 43 and the synthesized part patterns are displayed on the display unit 24.

Under such a condition that, as represented in FIG. 16C, the selected fighter 51a whose name is "A" is selected and his portrait is displayed, when the OK key 23 is manipulated at a step B5, the montage data "101", "202", . . . , "504" and "601" corresponding to this selected fighter 51a having his name of "A" are read out from the montage data register 39b. At the next step B6, these read montage data are allocated and stored into the basic fighting power data register 40a of the fighting power data RAM 40 as the life force data, the attacking force data, and the defensive force data for the respective parts.

At the next step B7, a total value of each of the life force data, the attacking power data, and the defensive force data of the selected fighter stored in the basic fighting power data register 40a is calculated by the CPU 31. Then, the first digit values of the respective total values are stored as the life force determining coefficient "A", the attacking force deter-

mining coefficient "B", and the defensive force determining coefficient "C" into the relevant coefficient registers 40b1 to 40b3 in the fighting power determining coefficient register 40b.

Thus, when the respective fighting power determining coefficients "A", "B", "C" could be obtained, calculations are performed for the life power data (ESP), the attacking power data (PK), and the defensive force data (TP) about the selected fighter 51a based on the life power determining calculation formula (formula 1), the attacking force determining calculation formula (formula 2), and the defensive force determining calculation formula (formula 3), which have been previously stored into the calculation formula ROM 34 at a step B8. The resultant data ESP, PK, TP are stored into the corresponding registers 40c1 to 40c3 in the determined fighting power data register 40c.

At the next step B9, as represented in FIG. 16D, these calculated data of the selected fighter 51a about the life force (ESP), the attacking force (PK), the defensive force (TP) are displayed in combination with the portrait of this selected fighter 51a on the liquid crystal display unit 24.

After the selected fighter by the user is registered in the above-described manner to set the fighter power data thereof, when it is judged at a step B10 that a lapse of 1 second is counted by a timer built in the CPU 31, the montage data corresponding to the first computer fighter is read from plural sets of the montage data previously stored in a montage ROM (not shown) for the computer fighter. Then, the various fighting power data ESP, PK, TP about this first computer fighter are calculated based on the read montage data in a similar manner to that of the above-described selected fighter at the subsequent step B11.

Thereafter, at a step B12, only the life force data (ESP) among the fighting power data of the computer fighter 51b is displayed on the liquid crystal display unit 24 in combination with the portrait of this computer fighter 51b synthesized based on the montage data corresponding to the first (LEVEL-1) computer fighter, as represented in FIG. 16E. Both the attacking force data (PK) and the defensive force data (TP) other than this life force data (ESP) are not displayed, but symbol data of "????" are alternatively displayed.

Thereafter, when at a step B13, it is so judged that a lapse of 1 second is counted by the timer built in the CPU 31, the random numbers of 0 to 9 are generated by the random number generating unit 41 at a step B14. Depending upon such a fact as to, for instance, whether the generated random number corresponds to the even number, or the odd number, a decision is made who is the first attacker for the above-described selected fighter 51a and computer fighter 51b.

At the next step B15, as shown in FIG. 16F, a life force representing bar 52a for the selected fighter 51a and a life force representing bar 52b for the computer fighter 51b, which indicate remaining amounts of the respective life force based on these life force data (ESP), are displayed in addition to the respective portraits of the fighters 51a and 51b. In correspondence with the above-described first attacker decision, an attacking arrow 53a indicative of the attacking direction between these fighters is displayed. In case of FIG. 16F, the attacking arrow 53a indicates that the selected fighter is the first attacker.

As described above, when the user selected fighter and the computer fighter are registered, such a condition that the battle game can be performed can be established.

BATTLE PROCESS OPERATION

A battle process operation for the montage battle gate as defined at the step X4 of FIG. 6 will now be described with

reference to flow charts shown in FIG. 12 to FIG. 14. At a first step C1 of the flow chart shown in FIG. 12, when it is judged that the game start key 20 is operated, the operation mode of the CPU 31 is set to the battle game mode. Thereafter, as shown in FIG. 16G, such a figure which represents the selected fighter under attacking is displayed at a step C2 in accordance with the bit map pattern previously stored in the ROM.

At the subsequent step C3, the random numbers "0" to "9" are produced from the random number generating unit 41. These random numbers are substituted for the random numbers of the attacking point calculation formula (formula 6) different from the basic value calculation formula (formula 5) previously stored in the attacking formula ROM 35, and the attacking point (BP) for the selected fighter 51a is calculated based upon this attacking point calculation formula (6).

On the other hand, in response to the above-explained process operation, the computer fighter 51b is attacked by the selected fighter 51a based on the calculated attacking point (BP) of the selected fighter 51a. Then, a step C4, new life force data (ESP) of the computer fighter 51b who has received the attack made by the selected fighter 51a by way of substituting the above-described attacking point (BP) value of the selected fighter 51a for another attacking point (BP) of the calculation formula (4) for the life force data after attacking which has been previously stored in the attacking formula ROM 35.

At the next step C5, as illustrated in FIG. 16H, there are displayed the portrait of the selected fighter 51a, the life force representing bar 52a corresponding to the attacking life force data (ESP), the portrait of the computer fighter 51b, and the life force representing bar 52b for the computer fighter, corresponding to the life force data (ESP) after attacking, which has been calculated at the previous step C4. Also, another attacking arrow 53b indicative of the attacking direction by the computer fighter 51b as the second fighter is displayed.

At the subsequent step C6, when it is judged that a predetermined time period has been measured by the CPU 31, such a figure that the computer fighter 51b is attacking is displayed in accordance with the bit map pattern previously stored in the ROM at a step C7 (see FIG. 16I).

Then, at a further step C8, the random numbers "0" to "9" are produced from the random number generating unit 41. These random numbers are substituted for the random numbers of the attacking point calculation formula (formula 6) different from the basic value calculation formula (formula 5) previously stored in the attacking formula ROM 35, and the attacking point (BP) for the computer fighter 51b is calculated based upon this attacking point calculation formula (6).

On the other hand, in response to the above-explained process operation, the selected fighter 51a is attacked by the computer fighter 51b based on the calculated attacking point (BP) of the computer fighter 51b. Then, a step C9, new life force data (ESP) of the selected fighter 51a who has received the attack made by the computer fighter 51b by way of substituting the above-described attacking point (BP) value of the computer fighter 51b for another attacking point (BP) of the calculation formula (4) for the life force data after attacking which has been previously stored in the attacking formula ROM 35. This calculated life force data (ESP) is stored into the life force data after attacking register 40d, and is updated as life force data (ESP) of the determined fighting power data register 40c.

At the next step C10, as illustrated in FIG. 16J, there are displayed the portrait of the selected fighter 51a, the life force representing bar 52a for the selected fighter, corresponding to the life force data (ESP) after attacking calculated at the step C9, the portrait of the computer fighter 51b, and the life force representing bar 52b for the computer fighter. Also, another attacking arrow 53b indicative of the attacking direction by the selected fighter 51a as the fighter is displayed.

Thereafter, at a step C11, when it is judged that the life force recovery key 22 is operated, the process operation is advanced to the life force recovery process operation (FIG. 15) in which the life force data (ESP) of the selected fighter 51a may be recovered.

Either after the recovery process operation of the life force data (ESP) has been performed, or while the life force recovery key 22 is not yet operated, until the value of the life force data (ESP) for either the selected fighter 51a, or the computer fighter 51b becomes "0", the above-described attacking operation between the selected fighter 51a and the computer fighter 51b as defined at the steps C2 to C10 is repeated (steps C13, C2 to C10).

On the other hand, when the value of the life force data (ESP) of the computer fighter 51b becomes, e.g., "0", while repeating the attacking operation between the respective fighters at the previous steps C2 to C10, as shown in FIG. 17A, the portrait of the selected fighter 51a and also the life force representing bar 52a are displayed. At the same time, the present portrait of the computer fighter 51b is substituted by another portrait indicative of loser's expression, which has been previously stored in the expression part pattern ROM 37. Furthermore, another life force representing bar 52a for the computer fighter corresponding to the value "0" of the above-described life force data is displayed (steps C13 to C14).

At the next step C15, the portrait of the first computer fighter (LEVEL-1) 51b who has lost this battle game is erased from the display screen, as illustrated in FIG. 17B. In this case, the process operation as defined from the step C14 to the step C15 is repeated, and then the images as shown in FIG. 17A and FIG. 17B are repeatedly displayed (steps C16 to C14 and C15).

When the OK key 23 is manipulated under such a condition that the fighter 51a selected by the user succeeds in the battle game and the first computer fighter (LEVEL-1) 51b loses this battle game, the montage data corresponding to the second computer fighter (LEVEL-2) 51c is read out from a plurality of montage data previously stored in the montage ROM (not shown) for the computer fighters. Based on the read montage data about the second computer fighter 51c, the fighting power data (ESP), (PK), (TP) are calculated. At the same time, as represented in FIG. 17C, another portrait of the second computer fighter (LEVEL-2) 51c which has been formed based upon the montage data corresponding to this second computer fighter (LEVEL-2) 51c, is displayed together with only the life force data (ESP) among these fighting power data thereof on the liquid crystal display unit 24 (steps C16 to C17).

At the next step C18, the attacking operation between the selected fighter 51a and the second computer fighter (LEVEL-2) 51c is performed in a similar manner to the above-described operation (step C18).

Conversely, when the value of the life force data (ESP) becomes, e.g., "0" while repeating the attacking operation between the first computer fighter 51b and the selected fighter 51a at the steps C2 to C10, both of the portrait of the

winner, i.e., the computer fighter 51b and the life force representing bar 52b are displayed as shown in FIG. 17E. Then, the present portrait of the selected fighter 51a is substituted by another portrait indicative of loser's expression, which has been previously stored in the expression part pattern ROM 37. Furthermore, the life force representing bar 52a for the selected fighter, corresponding to the value "0" of the life force data thereof is displayed (steps C13 to C19).

Thereafter, at a step C20, the present portrait of the selected fighter 51a who has lost this battle game is erased from the display screen of the liquid crystal dot matrix display unit 24.

LIFE FORCE RECOVERY PROCESS OPERATION

With reference to a flow chart shown in FIG. 15, the life force recovery process operation as defined at the step C12 of the battle game process operation shown in FIG. 12 and FIG. 13 will now be described in detail.

This life force recovery process operation is carried out when a judgement is made that the life force recovery key 22 is operated at a first step D1. In case that the operation mode of the CPU 31 is set to the life force recovery mode in accordance with this operation, as illustrated in FIG. 16K, such a sub-menu image is displayed on the liquid crystal display unit 24 based on the data about messages "ESP Item" and "Continue Fighting" read out from the battle message ROM 38 (step D2).

At the next step D3, it is judged that the "ESP item" to recover the life force is selected by operating the item selection key 16b, any one of the random numbers 0 to 9 is produced by the random number generating unit 41 (step D4).

Then, as illustrated in FIG. 16L, under such conditions that the portrait of the computer fighter 51b, the life force representing bar 52b of this computer fighter 51b, the life force representing bar 52a for the selected fighter, and the attacker's arrow 53a are represented, 10 sorts of icon patterns 54a to 54j previously stored in the life force recovery point ROM 36 are sequentially displayed in a predetermined rotation sequence from the icon corresponding to the random number generated at the previous step D4 (step D5).

When at the previous step D4, the random number produced from the random number generating unit 41 is, for instance, 4, the icon (4) 54d, stored in the life force recovery point ROM 36 is displayed and then the subsequent icon (5) 54e, icon (6) 54f, icon (7) 54g, . . . , are successively, rotatively displayed in an endless rail form (steps D6 to D5). It should be noted that the respective icons 54a to 54j correspond to such image data having different pictorial forms as animal image data.

At the next step D6, when it is judged that the OK key 23 is operated, and the icon presently displayed on the liquid crystal display unit 24 is, for example, the icon (3) 54c (see FIG. 16M), the life time recovery point of "27000" previously stored in the life time recovery point ROM 36 corresponding to the icon (3) 54c is read. Then, the icon 54c containing this read recovery point "27000" is stopped and displayed at the display point of the portrait for the computer fighter 51b whose life force is tried to be recovered at a step D7.

At the subsequent step D8, based on the life time recovery formula (7) previously stored in the calculation formula ROM 34, the previous value "27000" of the life force

recovery point corresponding to the icon 54c displayed in the step D7 is added to the value of the present life force data (ESP) stored in the determined fighting power data register 40c. The added value is stored in the life force data after recovery register 40e, and this value is used to update the latest life force data as new life force data (ESP) for the selected fighter. Then, the updated life force data is stored in the determined fighting power data register 40c (step D8).

As a consequence, the portrait of the selected fighter 51a is displayed instead of the above-described icon 54c (see FIG. 16N), and also the life force representing bar 52a corresponding to the life time data after recovery is displayed.

As a consequence, in accordance with the above-described montage battle game apparatus of the preferred embodiment, when the sub-menu image is displayed by operating the life force recovery key 22 and then the ESP item is selected so as to recover the life force of the relevant fighter (in this case, the selected fighter 51a), these icons 54a to 54j corresponding to 10 sorts of recovery points previously in the life force recovery point ROM 36, are successively displayed in the rotary mode in accordance with a predetermined sequence instead of the display of the selected fighter 51a whose life force should be recovered. Accordingly, when one of these icons (e.g., the icon 54c) is displayed, if the OK key 23 is operated, then the above-described rotary representation of this icon 54c is stopped. At the same time, since the value "27000" of the life force recovery point corresponding to this icon 54c is added to the value of the life force data owned by the present selected fighter 51a, this life force can be recovered in consideration of the user's desire. Moreover, the user can manually select one desirable icon from a plurality of icons used to recover the life force, which are successively displayed in the rotary mode, and then this user can consider the most favorable life force data, so that such a battle game with the higher grade can be realized.

MODIFICATIONS

Although the liquid crystal dot matrix display unit is employed as the means for announcing the comparison result in the above-described preferred embodiment, this comparison result may be informed by way of sounds, voice, or may be printed out.

Also, although the respective part numbers are employed as the portion data in the above-explained embodiment, the respective part pattern data per se for constituting the object image may be alternatively utilized.

Furthermore, as the object images for comparisons in the above-described embodiment, both the selected fighter 51a and the computer fighters 51b, 51c, . . . , are utilized which are selected from a plurality of formed montage images, and a plurality of computer fighters previously stored in the montage ROM for the computer fighters. Alternatively, the selected fighters may be fighted with each other, otherwise the computer fighters may be fighted with each other.

Additionally, the fighters are fighted within a single battle game apparatus in the above-explained embodiment. Alternatively, a plurality of apparatus equipped with communication functions are employed, while the comparison data are communicated with each other by way of radio waves, or lines. Then, the selected fighters may be fighted with each other on the different apparatuses.

What is claimed is:

1. An electronic game apparatus for electronically playing a competing game, comprising:

17

numeral value data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting two article images formed by combining portion images corresponding to respective portions of the articles;

point data storage means for storing therein first point data corresponding to the respective portion images for constituting each of said article images as to each of said two article images set by said setting means;

judging means for allocating third point data to each of said two article images based upon said first point data stored in said point data storage means in correspondence with said respective article images, and also second point data corresponding to said numeral value data generated from said numeral value data generating means, and for judging a competing result between said two article images set by said setting means based upon magnitudes of said third point data allocated to the respective article images; and

display means for displaying said competing result between said article images.

2. An electronic game apparatus as claimed in claim 1, wherein said display means includes means for displaying:

a first display in which said two articles set by said setting means are displayed as a first image; and

a second display in which at least one of said two article images displayed as said first display is changed into a second article image having a different shape from that of said at least one article image of said first image corresponding thereto in accordance with the judgement result made by said judging means;

said display means displaying said second article image as a visual image.

3. An electronic game apparatus for electronically playing a competing game, comprising:

numeral value data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles; and

judging means for judging a competing result (i) between a plurality of article images represented by said plurality of article image data set by said setting means based on both point data set in correspondence with portion image data for constituting said article image data, and point data corresponding to said numeral value data generated from said numeral data generating means; and (ii) between said article images judged by said judging means; and wherein:

said setting means includes:

portion image storage means for storing said plurality of portion image data corresponding to the respective portions of said articles;

designating means for designating one of said plural portion image data stored in said portion image storage means for each portion; and

article image data storage means for storing a plurality of article image data made by combining said portion image data with each other for the respective portions designated by said designating means.

4. An electronic game apparatus for electronically playing a competing game, comprising:

18

numeral value data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles; and

judging means for judging a competing result (i) between a plurality of article images represented by said plurality of article image data set by said setting means based on both point data set in correspondence with the portion image data for constituting said article image data, and point data corresponding to said numeral value data generated from said numeral data generating means; and (ii) between said article images judged by said judging means; and wherein:

said judging means includes:

article image data selecting means for selecting two sets of article image data from a plurality of article image data set by said setting means in order to perform the competing judgement; and

competing result judging means for judging the competing result between two sets of article images indicated by said selected article images based on both the point data corresponding to the numeral data generated by said numeral data generating means and the point data set in correspondence with said portion image data.

5. An electronic game apparatus as claimed in claim 1, wherein:

said two article images set by said setting means are face images.

6. An electronic game apparatus as claimed in claim 1, further comprising:

announcing means for announcing said competing result judged by said judging means in addition to a display of the competing result on said display means.

7. An electronic game apparatus for electronically playing a competing game, comprising:

numeral value data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles;

judging means for judging a competing result (i) between a plurality of article images represented by said plurality of article image data set by said setting means based on both point data set in correspondence with the portion image data for constituting said article image data, and point data corresponding to said numeral value data generated from said numeral data generating means; and (ii) between said article images judged by said judging means; and

image modifying means for modifying an article image represented by said article image data set by said setting means into another article image having a different form from that of the first-mentioned article image data in response to the competing result judged by said judging means.

8. An electronic game apparatus for electronically playing a competing game, comprising:

numeral value data generating means for generating a plurality of numeral data having different values in an irregular sequence;

setting means for setting a plurality of article image data constructed by combining portion image data with each other corresponding to respective portions of articles;

judging means for judging a competing result (i) between a plurality of article images represented by said plurality of article image data set by said setting means based on both point data set in correspondence with the portion image data for constituting said article image data, and point data corresponding to said numeral value data generated from said numeral data generating means; and (ii) between said article images judged by said judging means; and

image erasing means for erasing said article images represented by said article image data set by said setting means from said display means in accordance with said competing result judged by said judging means.

9. An electronic game apparatus for electrically playing a competing game, comprising:

display means for displaying at least an image;

numeral data generating means for generating a plurality of numerical data having different values in an irregular sequence;

point data storage means for storing a plurality of point data corresponding to a plurality of article image data, respectively;

image data storage means for storing said plurality of article image data constructed by combining portion image data with each other with respect to a plurality of portions;

selecting means for selecting at least two sets of article image data from said plurality of article image data stored in said image data storage means;

display control means for causing at least said two sets of article image data selected by said selecting means to be displayed on said display means;

game starting instructions means for instructing starting of said competing game;

reading means for reading first point data corresponding to each of said two article images displayed on said display means in response to the instruction issued from said game starting instruction means from said point data storage means for each of said two article image data; and

competing result determining means for allocating to said two sets of article image data, third point data formed based on both said first point data read by said reading means for each of said two sets of article image data and second point data corresponding to said irregularly generated numeral data, and for determining a competing result between said two article images based upon comparison results in magnitudes of said third point data allocated to said two sets of article image data.

10. An electronic game apparatus as claimed in claim 9, further comprising:

announcing means for announcing said competing result judged by said competing result determining means.

11. An electronic game apparatus as claimed in claim 10, further comprising:

image modifying means for modifying an article image represented by said article image data selected by said selecting means into another article image having a different form from that of the first-mentioned article image data in response to the competing result judged by said competing result determining means.

12. An electronic game apparatus as claimed in claim 9, further comprising:

image erasing means for erasing said article images represented by said article image data selected by said

selecting means from said display means in accordance with said competing result judged by said competing result determining means.

13. An electronic game apparatus as claimed in claim 9, further comprising:

partial image data storage means for storing a plurality of partial image data with respect to a plurality of portions;

selecting means for selecting said plurality of partial image data stored in said partial image data storage means with respect to each of said portions; and

image storage controlling means for causing a plurality of article image data constructed by combining the respective partial image data selected by said selecting means for the respective portions to be stored into said image data storage means.

14. An electronic game apparatus for determining a competing result based on at least first and second point data, comprising:

display means for displaying at least images representative of image data;

first data storage means for storing a plurality of first point data having different points corresponding to a plurality of first different image data and respective first image data;

second data storage means for storing second point data corresponding to a plurality of second image data constructed by combining partial image data corresponding to a plurality of image portions and said respective image data;

selecting means for selecting arbitrary second image data from said plurality of second image data stored in said second data storage means;

display control means for sequentially displaying said plurality of first different image data stored in said first data storage means and at the same time, for displaying the arbitrary second image data selected by said selecting means;

manual operation means for selecting arbitrary image data by a manual operation from said plurality of different image data sequentially displayed on said display means;

reading means for reading first point data corresponding to the first image data instructed by said manual operation means from said first data storage means, and also for reading second point data corresponding to the respective portion image data for constituting the second image data being displayed on said display means, while operating said manual operation means, from said second data storage means; and

competing result determining means or setting third point data defined by said first and second point data read from said reading means as one of said first and second point data, and for determining a competing result based on a comparison result in magnitudes of said first and second point data.

15. An electronic game apparatus as claimed in claim 14, further comprising:

announcing means for announcing said competing result determined by said competing result determining means.

16. An electronic game apparatus as claimed in claim 14, further comprising:

image modifying means for modifying an article image represented by said second arbitrary image data

selected by said selecting means into another image having a different form from that of said second arbitrary image data in response to the competing result determined by said competing result determining means.

17. An electronic game apparatus as claimed in claim **14**, further comprising:

image erasing means for erasing a second image represented by said second arbitrary image data selected by said selecting means from said display means in accordance with said competing result determined by said competing result determining means.

18. An electronic game apparatus as claimed in claim **14**, further comprising:

partial image data storage means for storing a plurality of partial image data with respect to a plurality of portions;

selecting means for selecting said plurality of partial image data stored in said partial image data storage means with respect to each of said portions; and

image storage controlling means for causing a plurality of article image data constructed of combining the respective partial image data selected by said selecting means for the respective portions to be stored into said image storage means, as said second image data.

19. An electronic game apparatus as claimed in claim **14**, further comprising:

article image data storage means for storing a plurality of article image data;

selecting means for selecting at least two sets of article image data from said plural image data stored in said article image data storage means; and

image data storage controlling means for causing said two sets of article image data selected by said selecting means to be stored into said second data storage means as said second image data.

20. An electronic game apparatus as claimed in claim **14** wherein one image data among said plural second image data stored in said second data storage means corresponds to article image data produced by a user by combining the partial image data with each other with regard to said plurality of portions, whereas the other image data among said second image data corresponds to article image data produced by a preselected combination of said partial image data with respect to said plurality of portions.

21. An electronic game apparatus as claimed in claim **14** wherein said display control means performs such a control operation that said plurality of first image data stored in said first data storage means are sequentially displayed on said display means every time a predetermined time has passed.

22. An electronic game apparatus as claimed in claim **14** wherein said plurality of second image data selected by said selecting means correspond to face image data constructed by combining the partial image data for said plural portions.

* * * * *

United States Patent [19]

Hallman, Jr. et al.

[11] Patent Number: **5,159,549**

[45] Date of Patent: **Oct. 27, 1992**

[54] **MULTIPLE PLAYER GAME DATA PROCESSING SYSTEM WITH WAGER ACCOUNTING**

[75] Inventors: **John R. Hallman, Jr.,** Columbia; **Richard A. Schrenker,** Baltimore, both of Md.

[73] Assignee: **Poker Pot, Inc.,** Baltimore, Md.

[21] Appl. No.: **39,778**

[22] Filed: **Apr. 16, 1987**

Related U.S. Application Data

[63] Continuation of Ser. No. 616,291, Jun. 1, 1984, abandoned.

[51] Int. Cl.⁵ **G06F 15/44; G06F 15/28**
[52] U.S. Cl. .... **364/412; 273/138 A**
[58] Field of Search ... 364/200 MS File, 900 MS File, 364/412; 273/85 CP, 85 G, 237, 274, DIG. 28, 138 A, 1 E; 340/323 R

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,420,526	1/1969	Berger	273/148 R
3,810,627	5/1974	Levy	273/138 A
3,819,186	6/1974	Hinterstocker	273/138 A
3,876,208	4/1975	Wachtler et al.	273/138 A
4,030,764	6/1977	Mattos	273/148 A
4,052,073	10/1977	Miller	273/148 R
4,093,215	6/1978	Ballard	273/1 E
4,193,600	3/1980	Armstrong et al.	273/148 R
4,322,612	3/1982	Lange	364/412 X
4,339,798	7/1982	Hedges et al.	364/412
4,373,719	2/1983	Nelson et al.	273/1 E
4,467,424	8/1984	Hedges et al.	364/412
4,760,527	7/1988	Sidley	364/412

FOREIGN PATENT DOCUMENTS

2449631	10/1974	Fed. Rep. of Germany .	
2067080	7/1981	United Kingdom	273/85 G

OTHER PUBLICATIONS

Prisament, Steve, "Poker May Get a New Deal" *Venture,* Atlantic City, N.J., Jul. 20, 1984, V1, V3.
Findler, N. V., "Computer Poker", *Scientific American,* vol. 239, No. 1, Jul. 1978, 144–51.

Primary Examiner—Robert Weinhardt
Attorney, Agent, or Firm—Leitner, Greene & Christensen

[57] **ABSTRACT**

A data processing system is provided for tallying wealth accumulation among a plurality of competing players. Each player has a game entry device coupled to a central processing unit. The CPU receives data on an interrupt basis from each of the player stations and regulates the ordered play among the competitors. The CPU is responsive to the data for indicating a winner, calculating the accumulated point total or wealth of each of the players and for indicating the amount necessary for a player to risk in order to stay in the competition. Anyone of the player stations designated may perform house or banking functions in addition to player functions.

6 Claims, 16 Drawing Sheets

FIG.I

FIG 2

FIG. 3A

FIG.3B

FIG.3C

FIG.4A
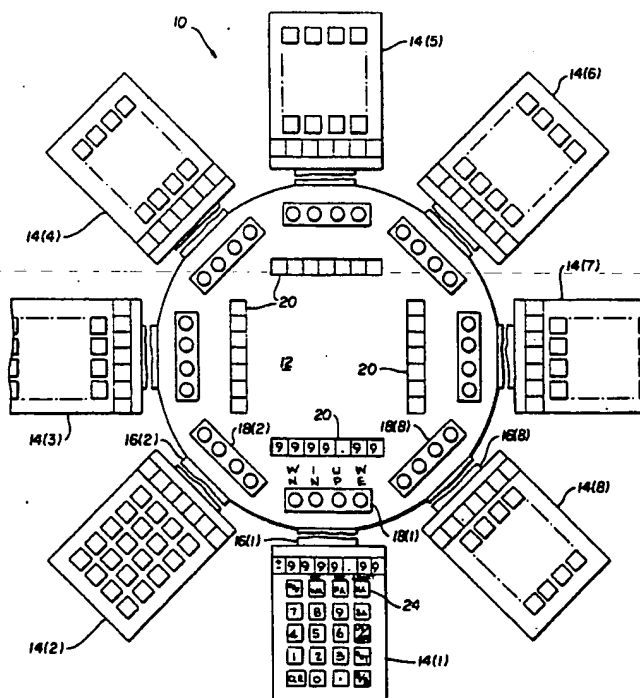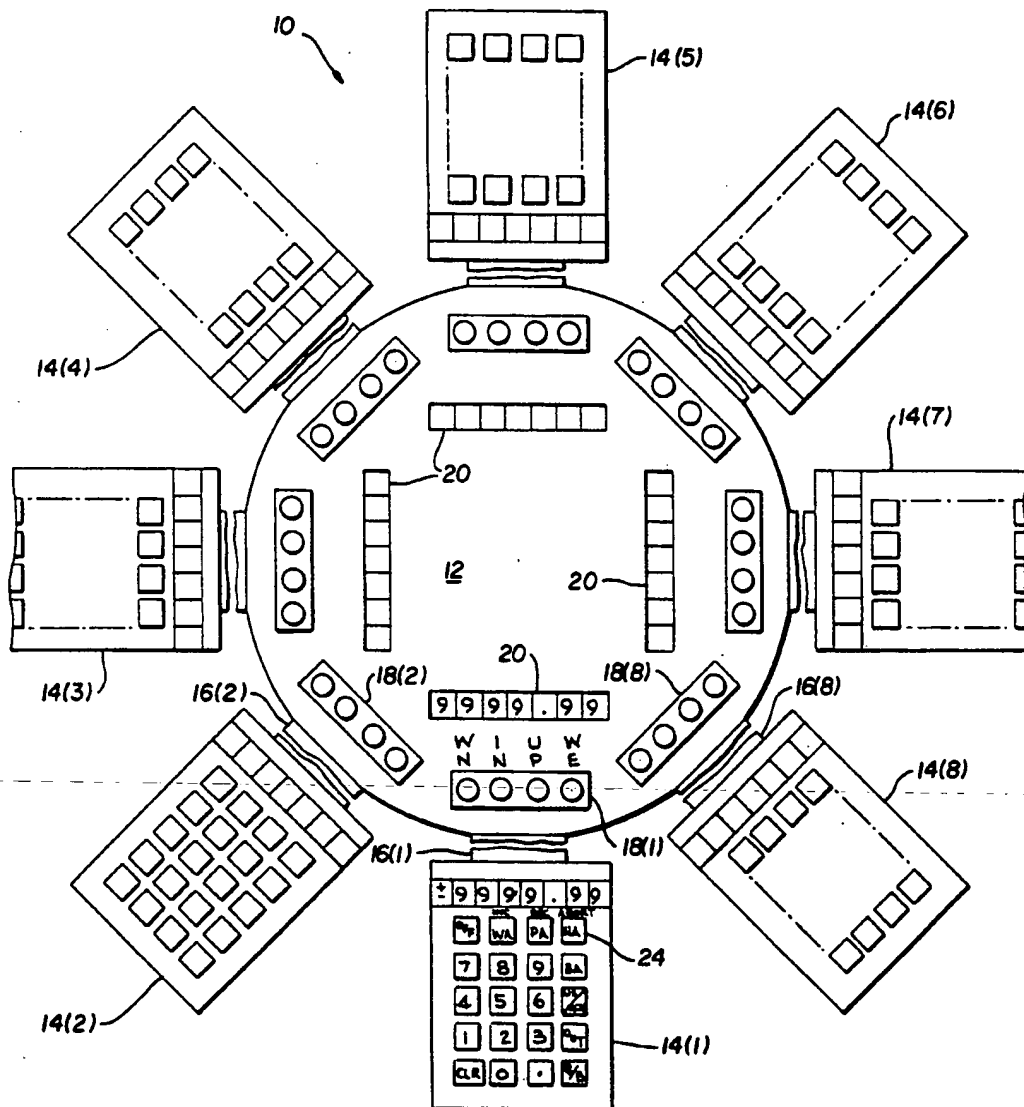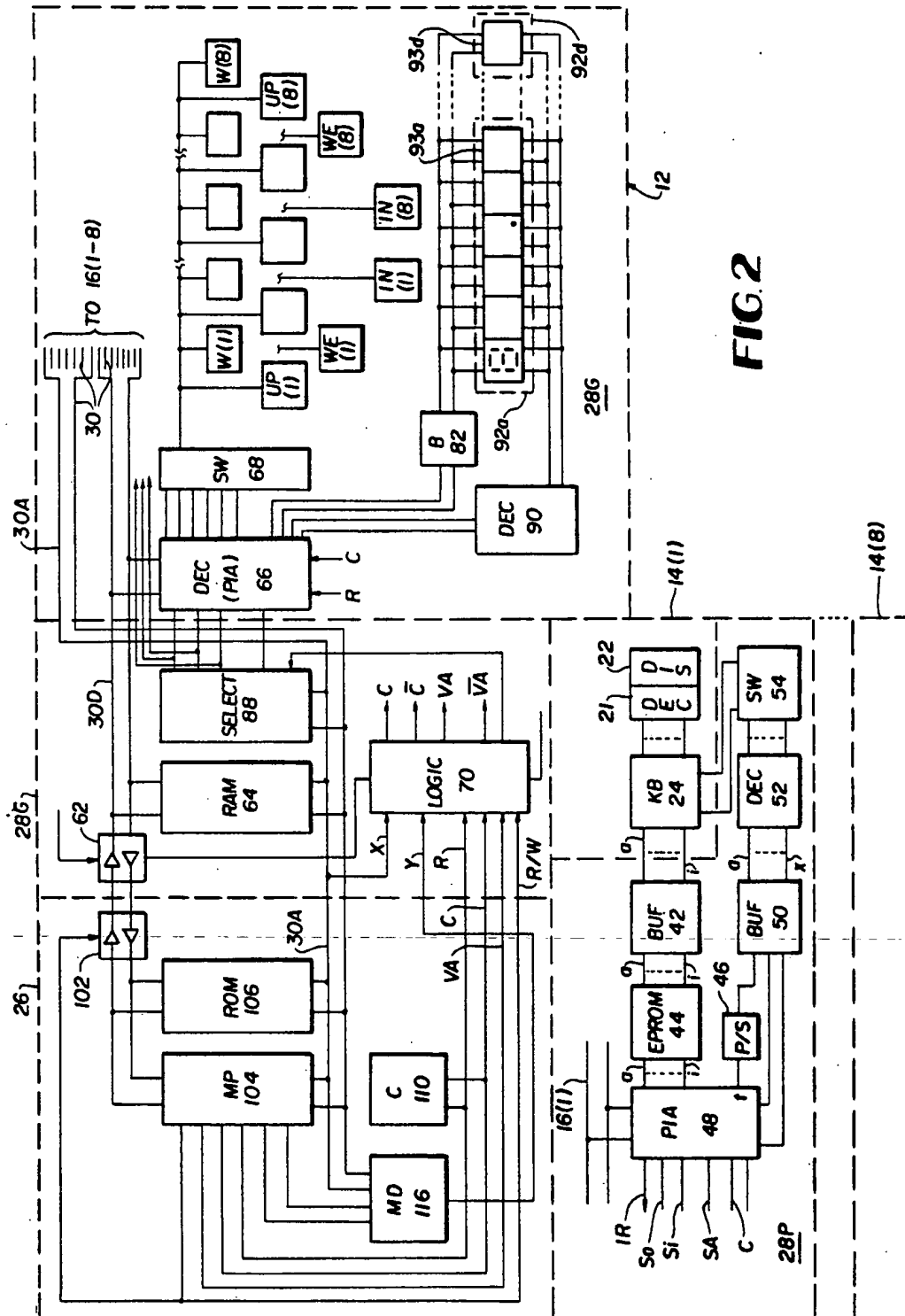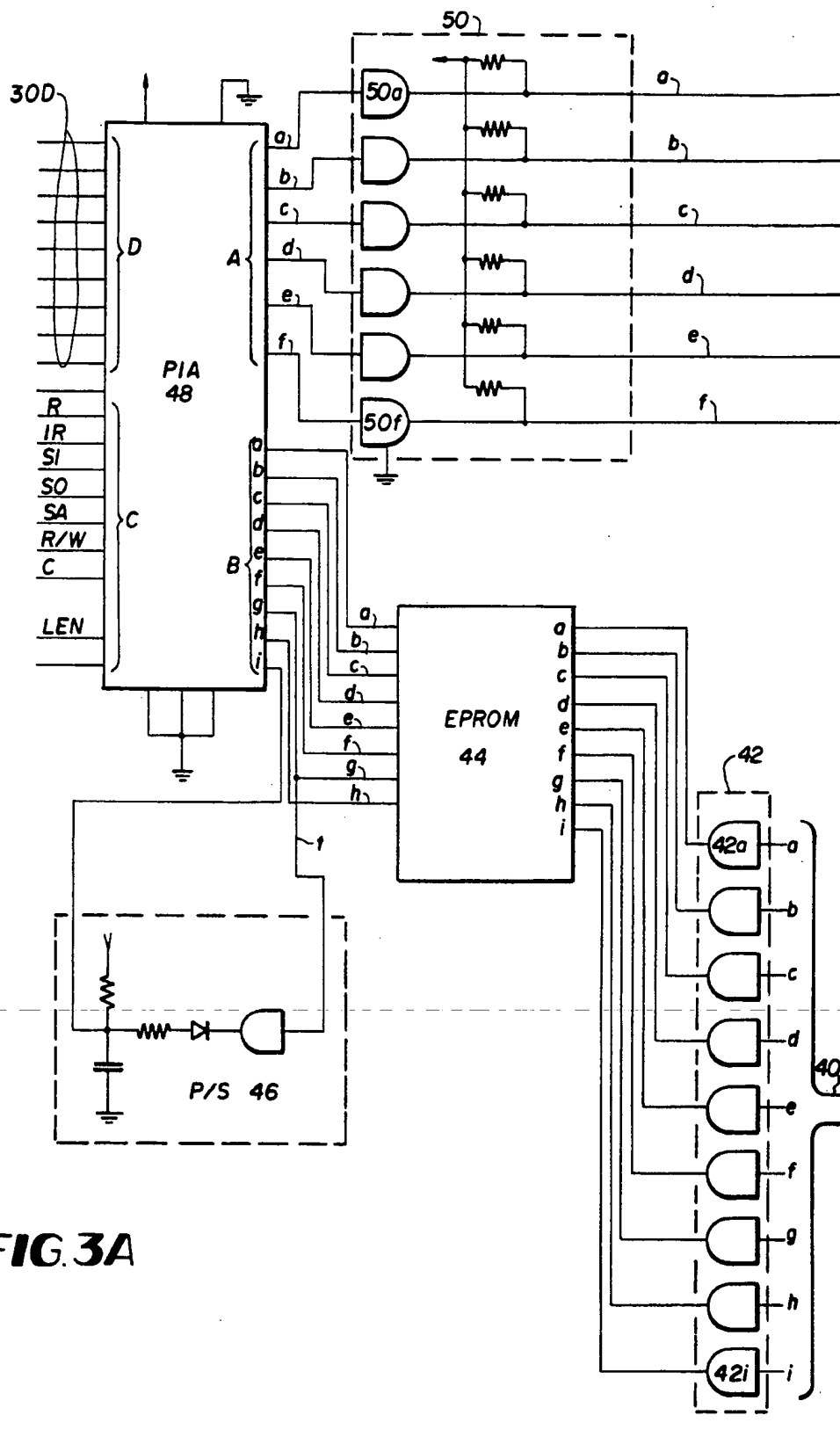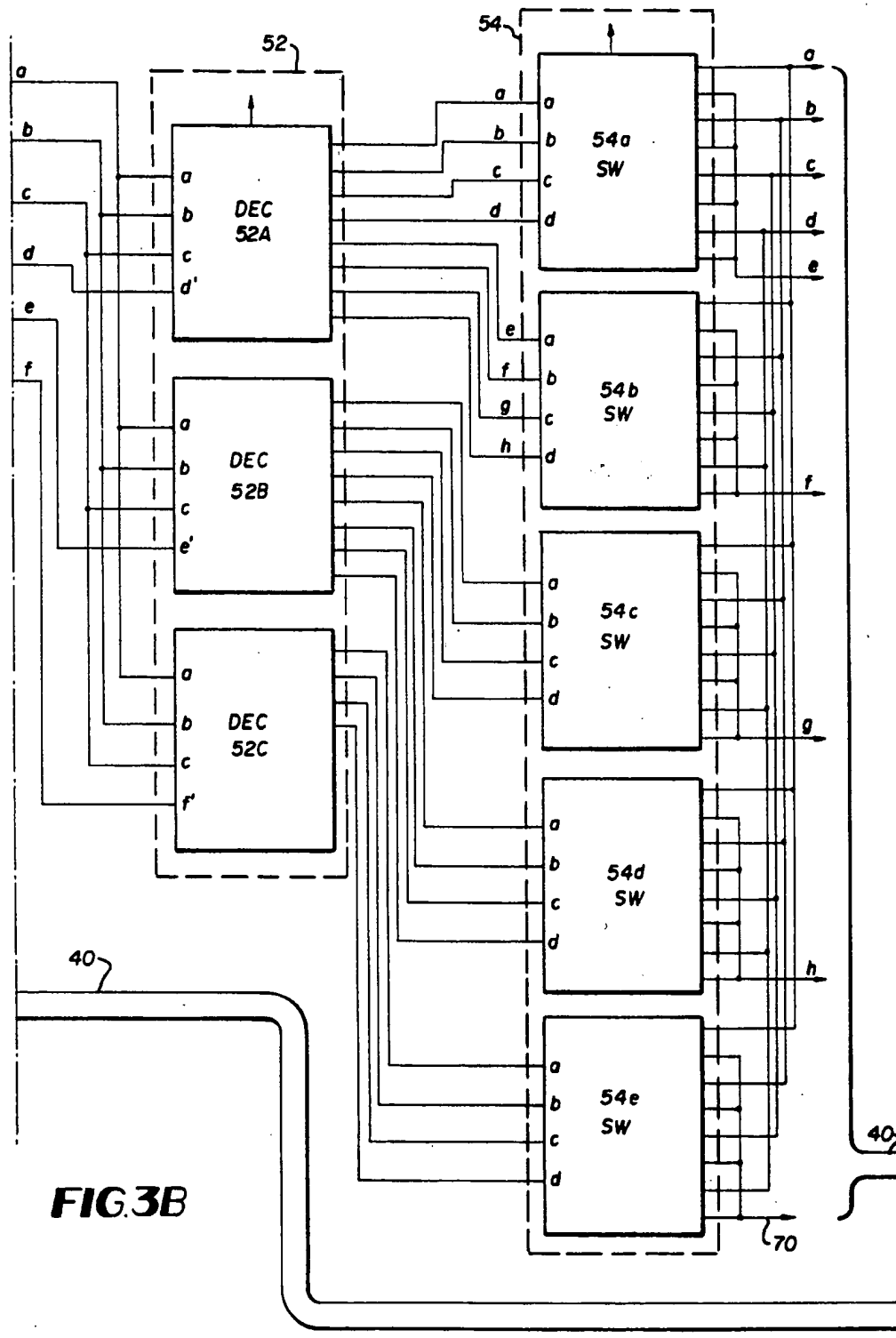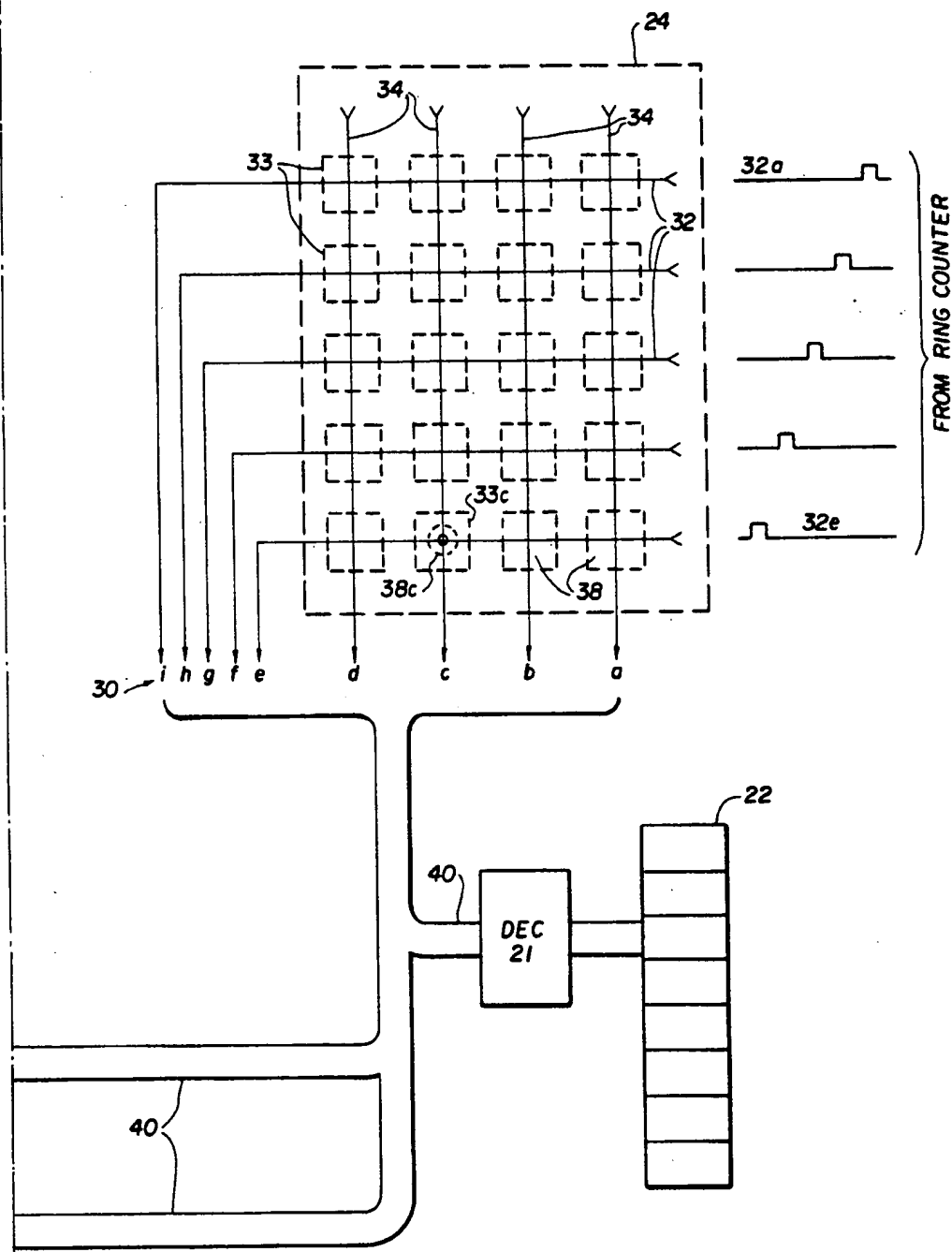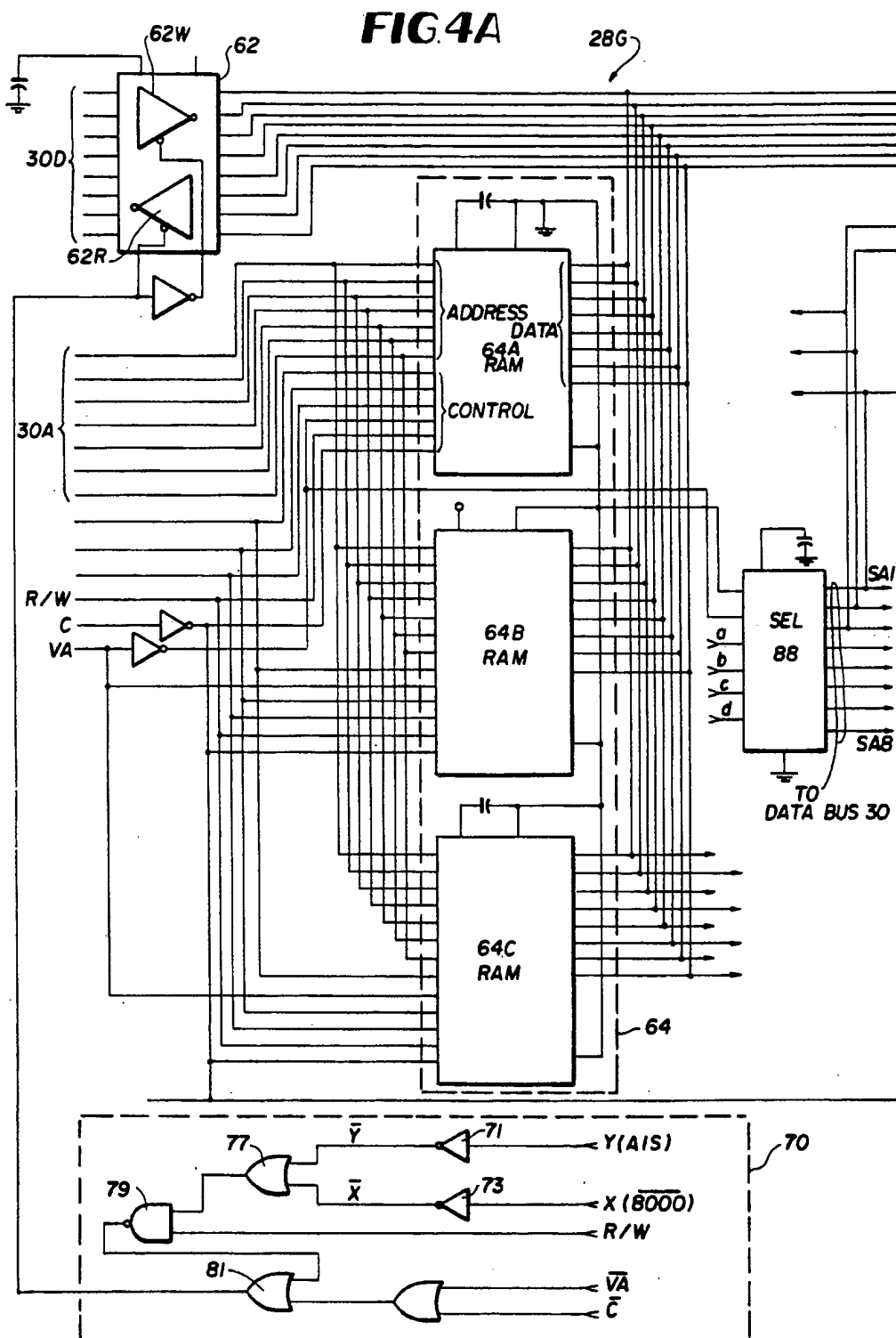
FIG.4B

FIG.4C

FIG.5A

FIG.5B
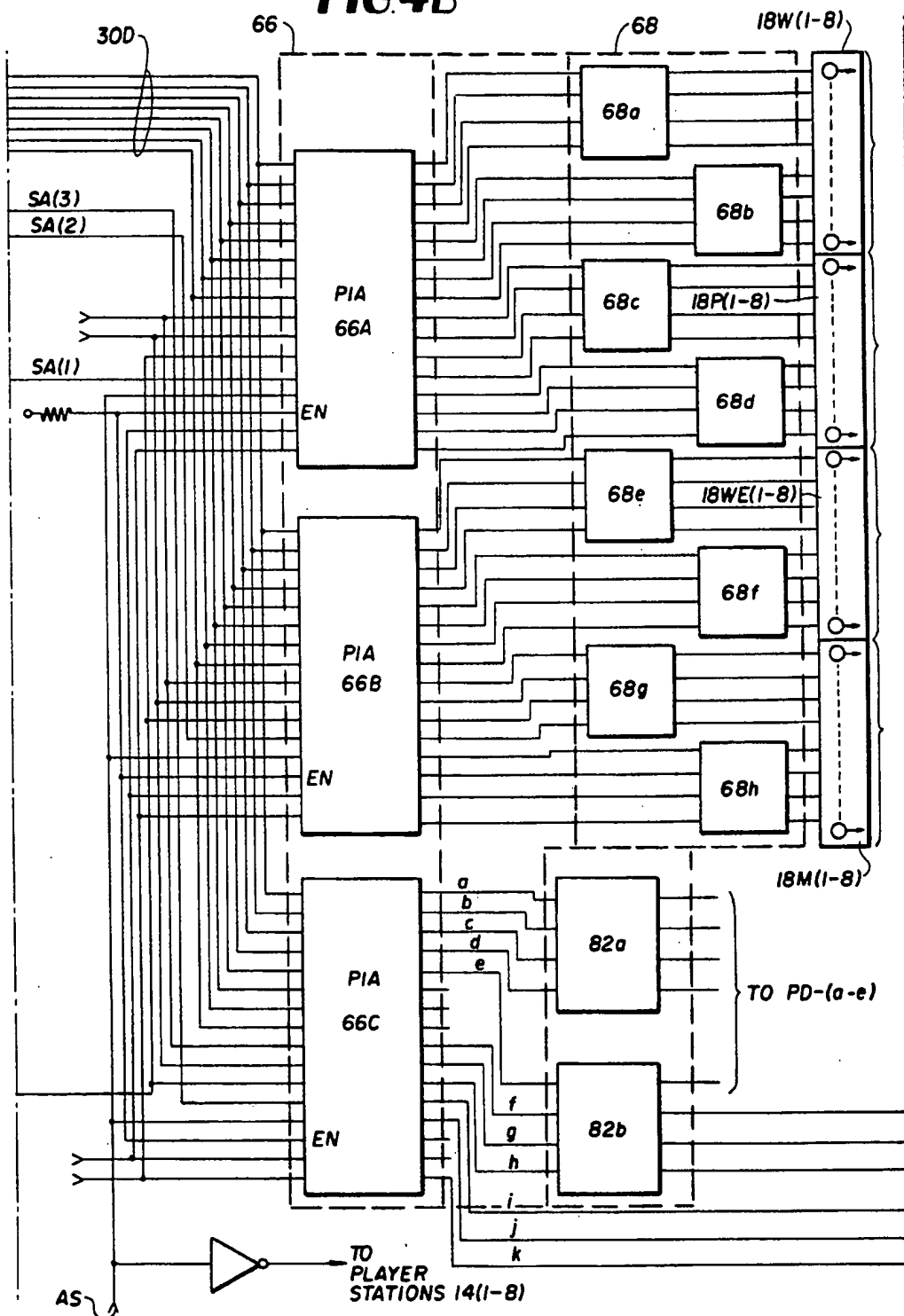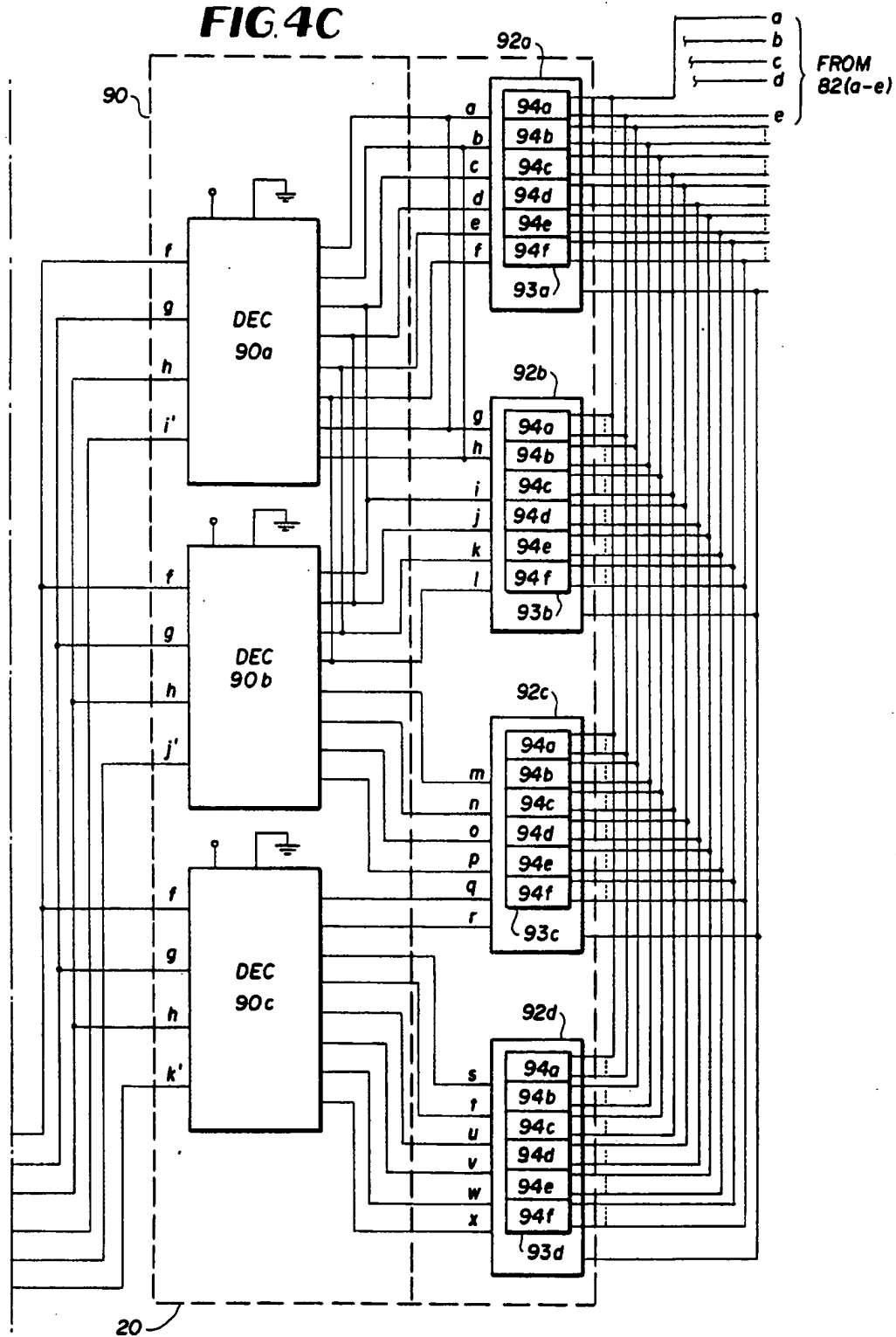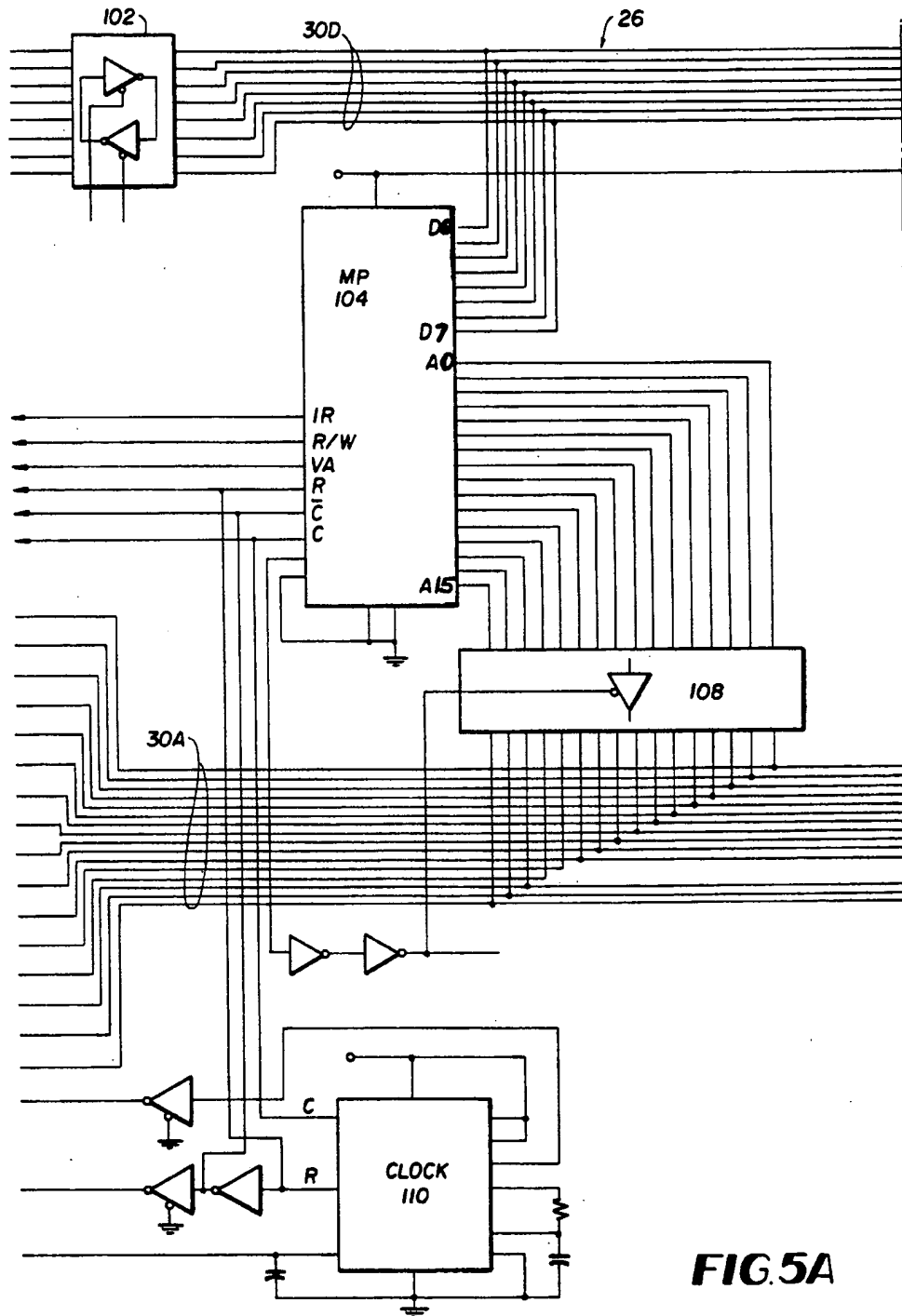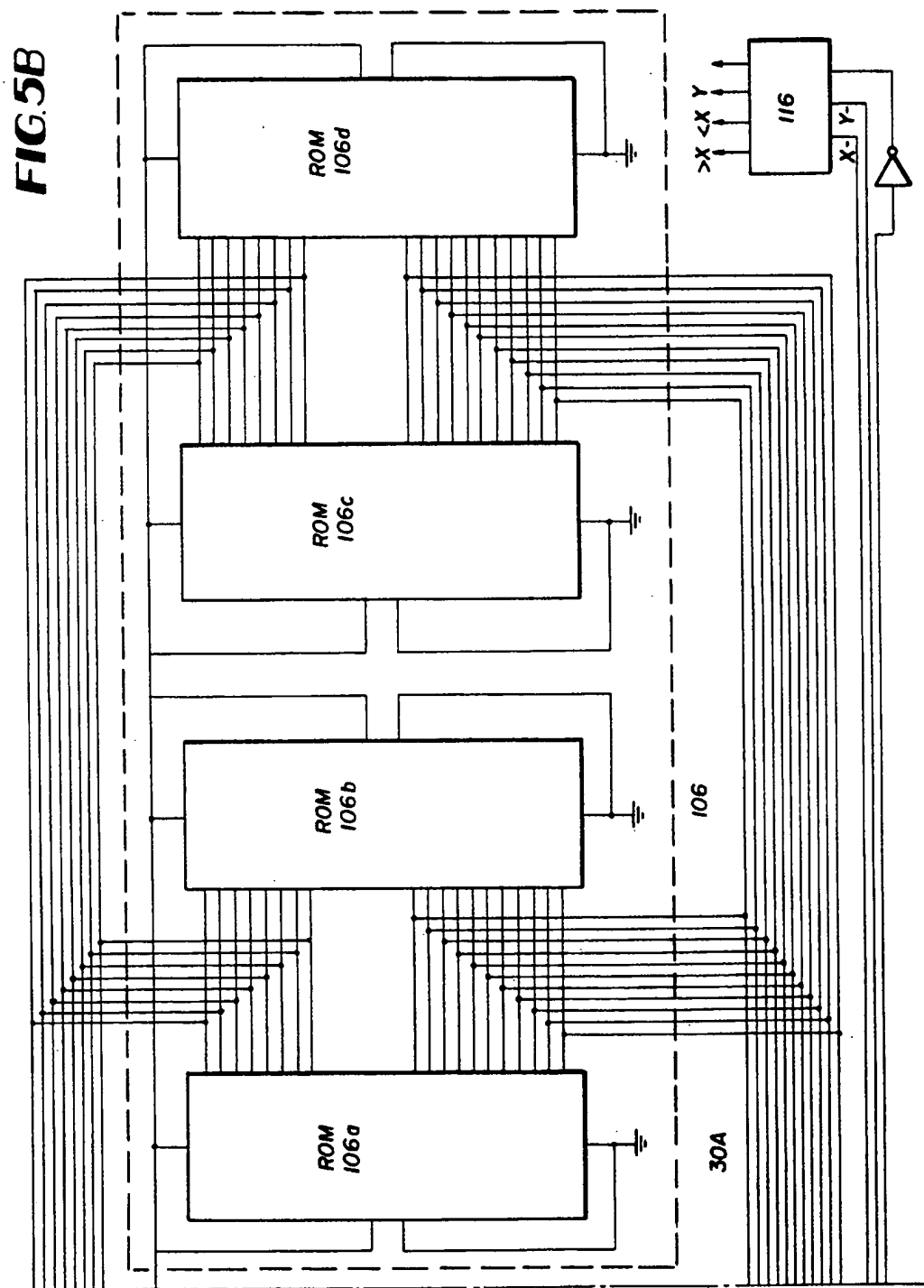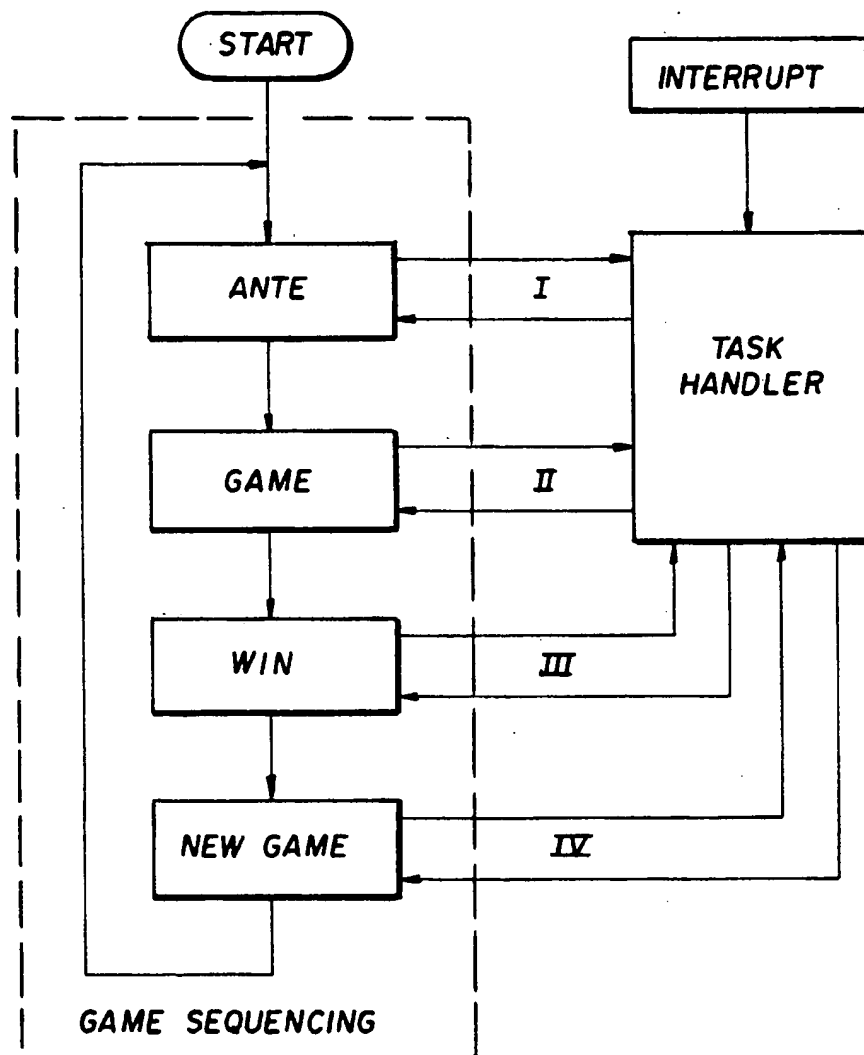
FIG.6

FIG.7A

FIG.7B

FIG.8A

FIG.8B

COMPUTE ACCOUNTS — 302

PLAYER STATUS — 304

INCREMENT GAMA STA No.(GSN) — 306

NEW GSN = ? — 308

SET NEW GSN = 1 — 316

NEW GSN = OLD GSN — 312

SET NEXT PLAYER UP — 314

CALCULATE & DISPLAY BA — 316

ONLY ONE PLAYER INDICATE WIN & DISPLAY AMT — 318

GO TO WIN ROUTINE — 346W

RETURN VIA WIN MODE — 322

CLEAR SB — 214

FIG.8C

FIG.8D

1
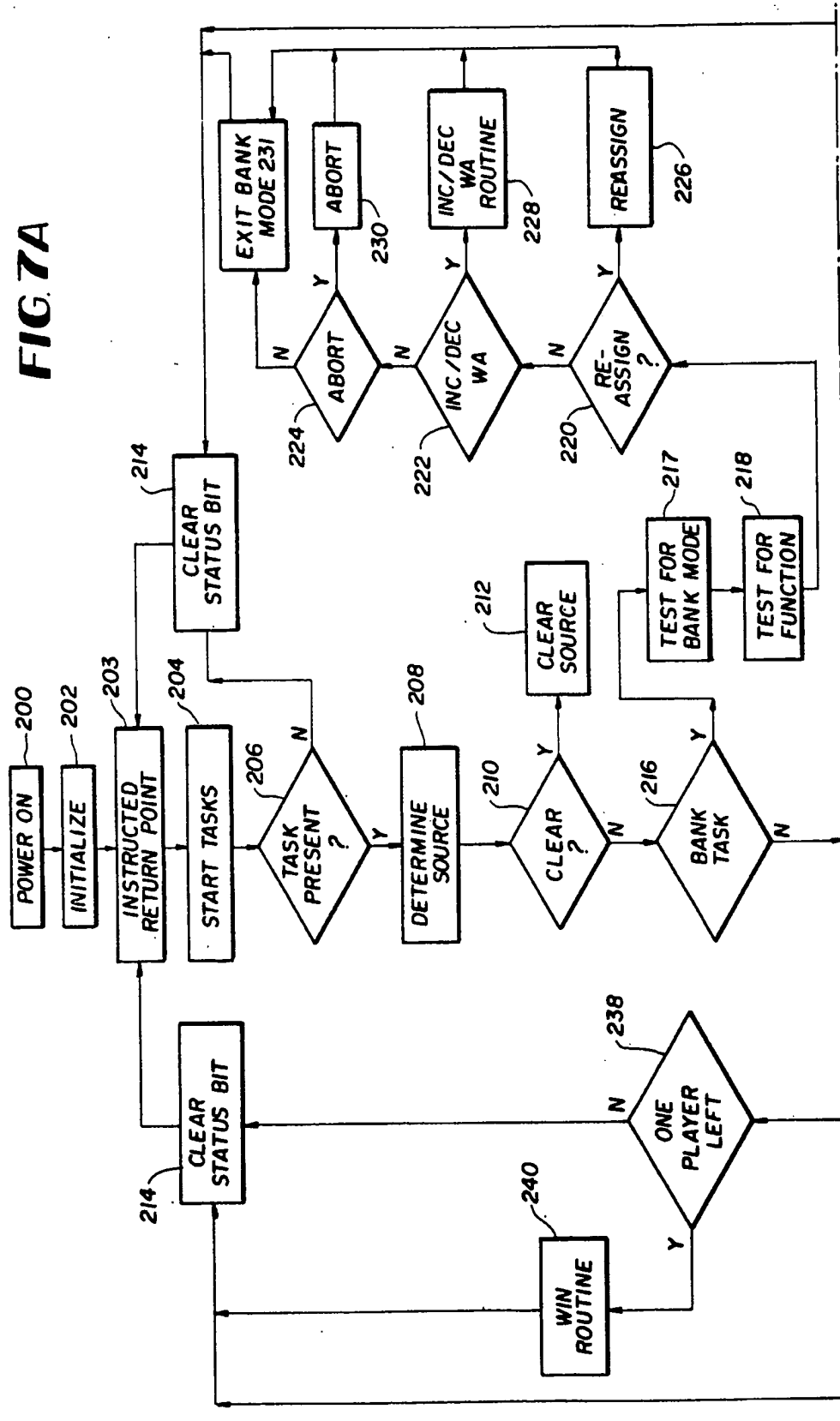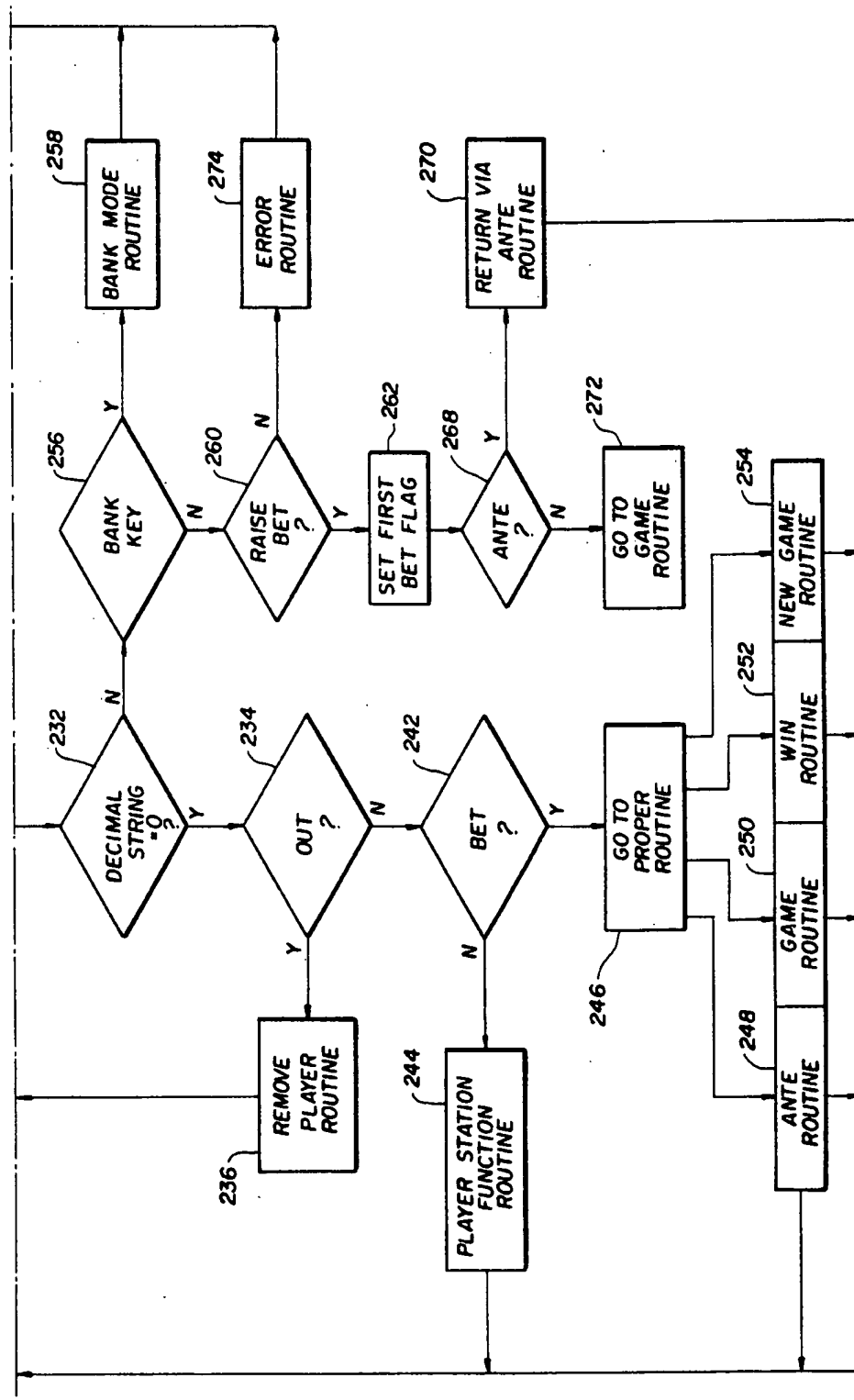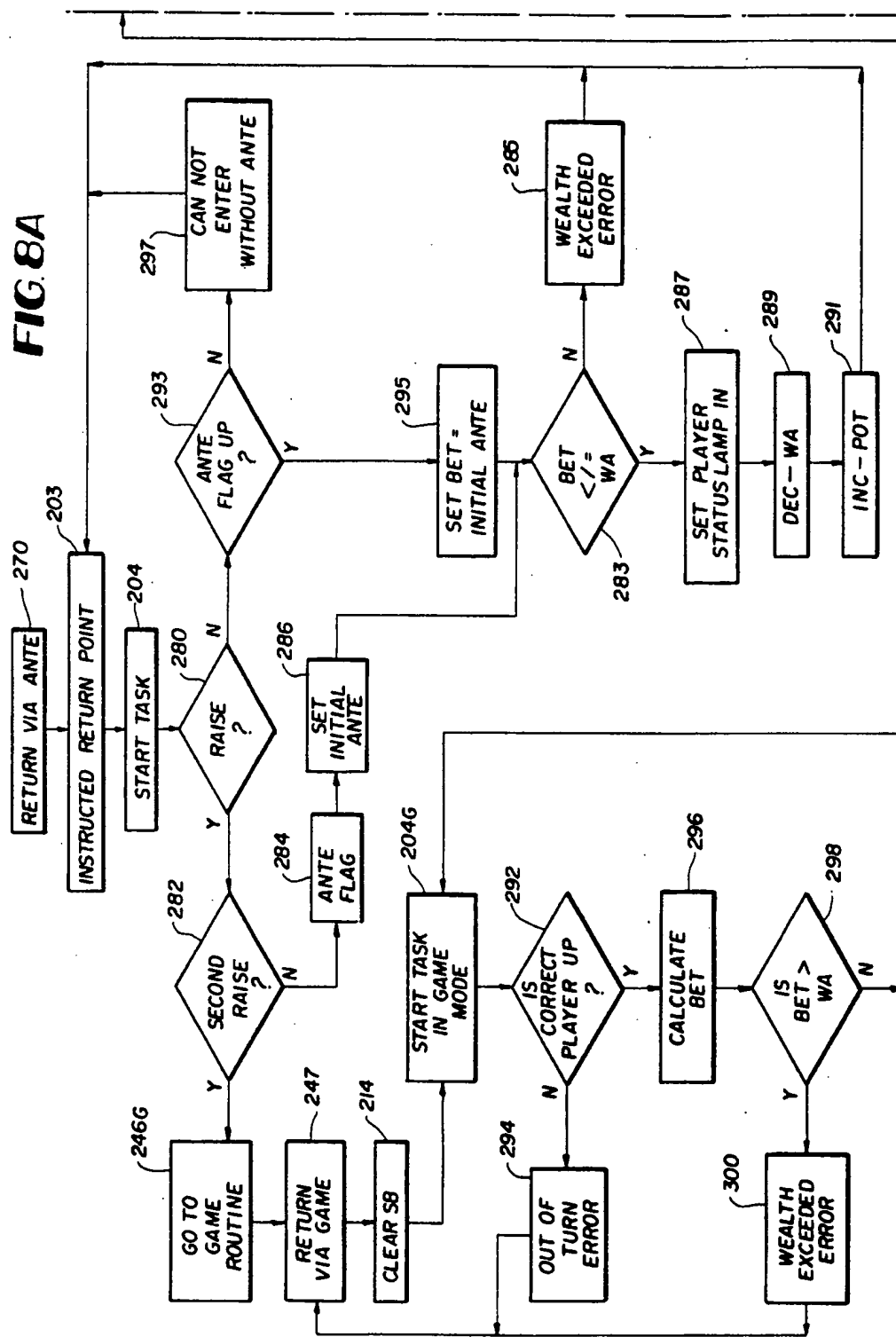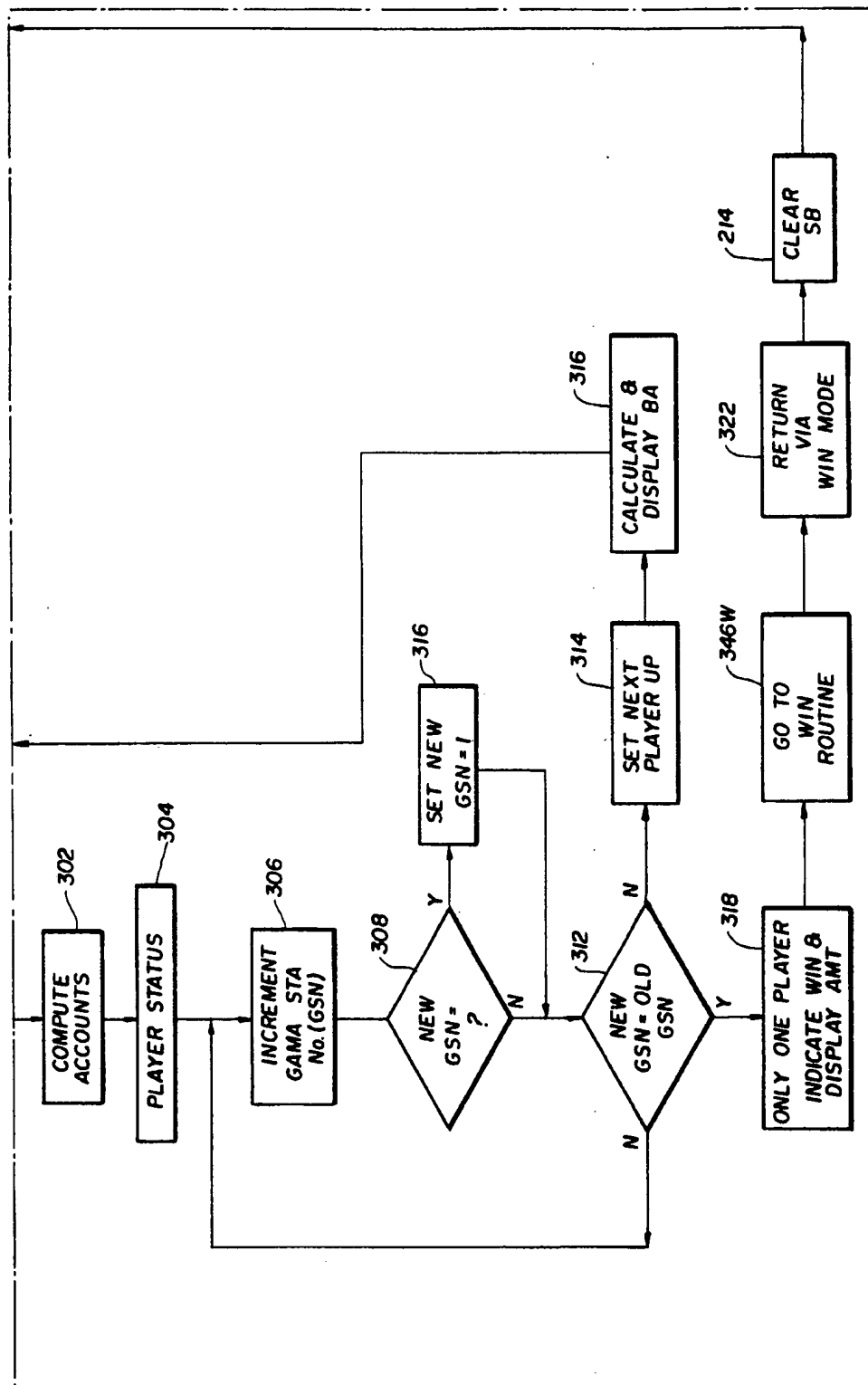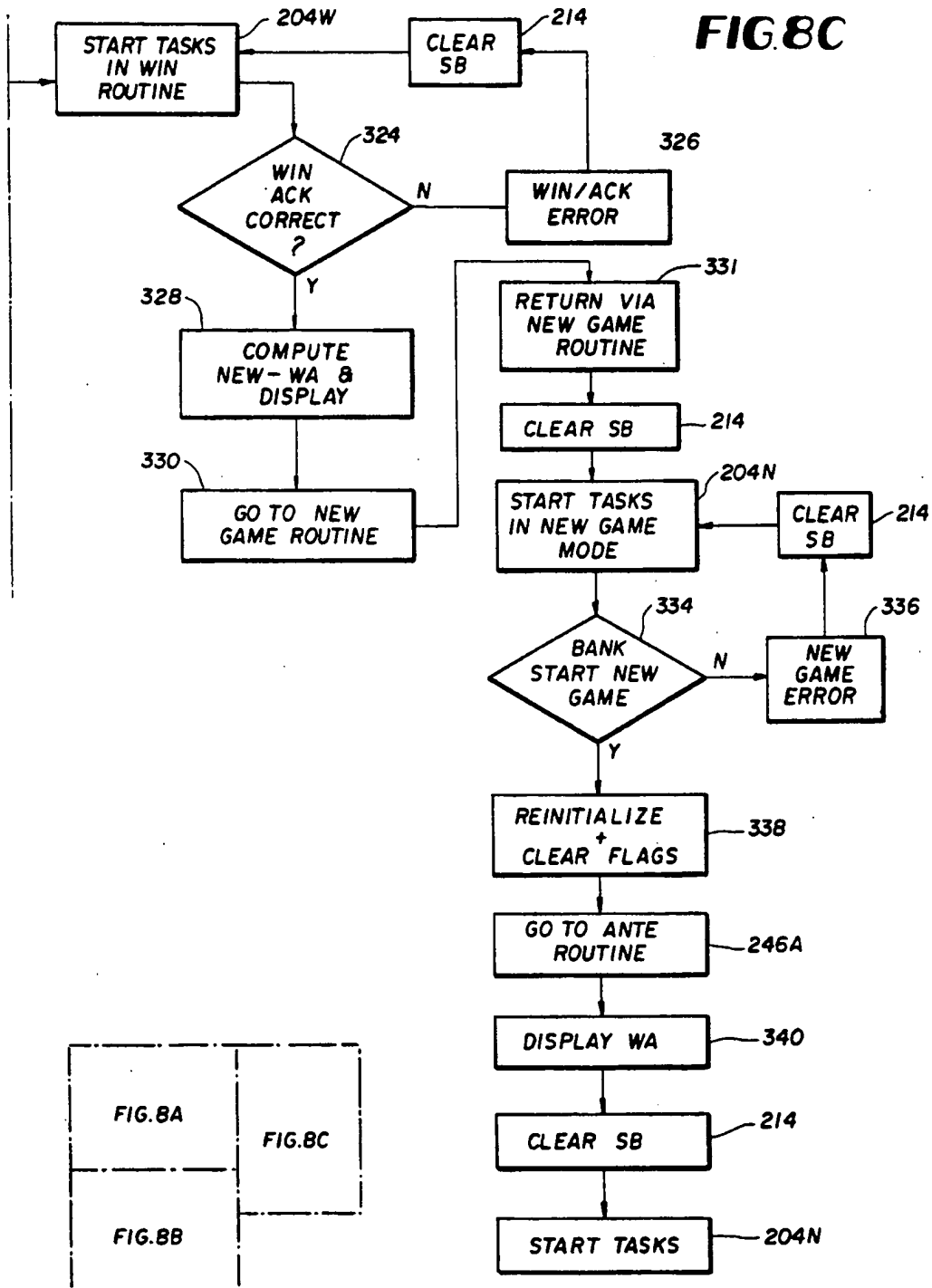
MULTIPLE PLAYER GAME DATA PROCESSING SYSTEM WITH WAGER ACCOUNTING

This is a continuation of application Ser. No. 616,291 filed Jun. 1, 1984, now abandoned.

BACKGROUND OF THE INVENTION

The invention relates to an multiple player game data processing system with wager accounting for keeping track of holdings, winnings, or accumulated points among a plurality of players. Particularly, in a preferred embodiment, the invention is used as a tallying, game and player sequencing device in a game of poker.

In the game of poker, two or more players receive cards and bet or wager against each other in accordance with known rules. Bets are tracked by use of cash or colored chips, each representing points scored or a denomination of money. A player buys chips from the bank or house and plays with chips as the equivalent of money in making wagers during the game or games. At the end of play, the player cashes the chips for money. The allotment and cashing of chips is time consuming, susceptible to error and inconvenient.

Sometimes during a game, betting errors occur. For example, it is difficult to keep track of which players are in or out of the game and for what wager amount, especially when there are multiple raises. There are also other inconveniences associated with chips or cash. For example, table space is occupied, chips must be stacked after each transaction and chip stacks are sometimes upset and must be re-stacked.

The present invention eliminates the foregoing difficulties and, in addition, adds excitement and a new strategic dimension to the game of poker. For example, the invention displays for all players the pot at stake in a particular game. The wagered amount the raise, and the amount necessary for a player to stay in may be displayed to an individual player upon demand. It also displays which players are in the game, whether a bet exceeds a player's wealth, whose turn it is to be, and the winner at the completion of a hand. The order of betting is strictly enforced, except during the ante phase when random betting and entry is permitted.

In addition, individual players have private access to data indicative of their personal total wealth and their personal stake in each individual on-going game. The game is accellerated for additional excitement because play is not seriously interrupted for the banking tasks and annoying pot and game status inquiries of inattentive players.

SUMMARY OF THE INVENTION

There has provided a multiple player game data processing system with wager accounting for determining respective aggregate points or wealth, gains and losses and the stake of each of a plurality of players wagering against each other in a game of chance. The system includes means for determining an aggregate amount, or pot, wagered by the players, means for determining the amount necessary for a player to enter and remain in the game as initial and successive wagers are made, and means for increasing and decreasing each player's respective aggregate wealth or accumulated points in accordance with the wagers made and the outcome of the game. The system includes a player game entry device or station for each player including respective wealth acknowledgement means for acknowledging

2

and receiving wealth upon demand, wager selector means for registering and indicating an amount wagered and withdrawal selector means for registering withdrawal from the game. Player wealth inquiry means for each respective player station is operative only at such station for selectively indicating a wealth amount for the respective player. A central processing unit is operatively coupled to the player stations for receiving the respective wagers and computing and indicating the pot at stake; for reducing such player's registered wealth in accordance with that player's respective wager; and for comparing the amount of each player's registered wager with a successive higher wager, for calculating the difference therebetween and for registering and indicating an amount necessary for such player to wager in order to remain in the game. Means at the central station is operative to register and indicate a win and the amount thereof in response to the operation of the withdrawal selector means by all of the players but the winner of said game. The wealth selector means communicates with the central station and the winner's player station for increasing the wealth amount of the winner by the pot amount registered by the central station in response to wealth acknowledgement by the winning player.

Means is provided for designating one player station with bank function, and for changing said designation in response to a signal from such station designating another as the bank and an acknowledgement from said station.

DESCRIPTION OF THE DRAWING

FIG. 1 is a schematic block diagram of the gaming device of the present invention showing eight player stations and a central station.

FIG. 2 is a schematic illustration of the layout of the system including a player station, interfaces and a central processing unit.

FIG. 3A–3C taken together from left to right make up a schematic diagram of a typical player station.

FIGS. 4A–4C taken together from left to right make up a schematic diagram of an interface between each of the player stations and the central processing unit showing inputs and outputs, along with system RAM and lamp and display drives.

FIGS. 5A–5B taken together from left to right make up an electrical schematic of the central processing unit with inputs and outputs.

FIG. 6 is a block diagram illustrating game and flow chart sequencing for various game phases of the present invention.

FIGS. 7A–7B from top to bottom make up a flow chart illustrating program functions of the gaming device of the present invention.

FIGS. 8A–8C make up a flow chart illustrating more details of program functions set forth in FIGS. 7A–7B.

FIG. 8D is a chart showing the arrangement of drawings for FIGS. 8A–8C.

DESCRIPTION OF THE PREFERRED EMBODIMENT

In FIG. 1 there is shown a gaming device 10 of the present invention. The preferred device 10 is, as hereinbefore described, an electrical system for keeping track of the various aspects of an on-going poker game among a plurality of players. The game comprises a master or central station 12 and a plurality of, preferably eight (8), player stations 14 (1–8) interconnected to the master

3

station 12 over dedicated channels 16 (1-8). Although eight player stations 14 (1-8) are shown, in a preferred embodiment, fewer may be used in a game. Suffixes (1-8) refer to particular player stations and are not used when referring to devices generally.

The master station 12 has a plurality of sets of indicators 18 (1-8), one for each player station 14. Each display 18 is dedicated to give information relevant to a player sitting opposite the same. Each set of indicators 18 includes a plurality of colored lights respectively indicating: WIN (red), IN/OUT (green), PLAYER UP (white), and WEALTH EXCEEDED (amber). In addition, a plurality of digital displays 20 are provided for giving numerical information. The indicators 18 and displays 20 are conveniently located so that each player can see the information conveyed by the master station 12. Although 4 digital displays 20 are shown, more or less may be used if desired because, unlike the indicators 18 (1-8), the digital displays 20 show information common to the game rather than individual players.

Each player station 14 has an alpha numeric keyboard 24 having individual keys 33 for inputting information and functional tasks to the system. The keys 33 are labeled or coded as follows:

Key Name	Function
0-9	Numerical inputs
(.)	Decimal Points
BET/RAISE/WIN ACK and NEW GAME	Bet, raise, win acknowledge and new game
OUT	Withdraw from game
INC WA	Increase Wealth Account
DEC WA	Decrease Wealth Account
BA	Display Bet Account
PA	Display Personal Account
WA	Display Wealth Account
HA	Display Hand Account
CLEAR	Clear Display
ABORT	Exit House/Bank Mode
OS/ACK	Initiate Bank move and Acknowledge Bank move

The function of the keys 33 and corresponding operation of the device 10 shall be hereinafter described in detail in conjunction with a description of the various functional elements of the device.

System Operation

Referring to FIG. 2, there is shown in block form the central station 12 incorporating therein a central processing unit (CPU) 26 and a general interface 28G. There is also shown in detail one of the eight player stations 14 (1) connected in parallel to the central station 12 over the corresponding channel 16 (1). Each player station 14 has a player station interface 28P which couples the keyboard 24 of the player station 14 with the CPU 26 via the general interface 28G and the respective dedicated channel 16.

In FIG. 2, the CPU 26 is coupled to the general interface 28G over data bus 30D and address bus 30A. Each player station 14 is coupled in parallel to the data bus 30D of the central station general interface 28G via a dedicated channel 16. A station address line SA, hereinafter described, addresses each player station 14 by a dedicated code unique to such station. The CPU thus communicates with each player station 14 individually and exclusively.

4

Player Station 14

For a description of the player station 14, reference is directed to FIGS. 2 and 3A-C. Each player station 14 comprises a keyboard 24 with decoder 21 and display 22 and a player station interface 28P. One such keyboard 24, with decoder 21 and display 22 is incorporated into a hand-held calculator (not shown) such as Model No. TI-1000 manufactured by Texas Instruments. The device accepts inputs by mechanically shorting a matrix of respective horizontal and vertical wires 32 and 34. In FIG. 3C, such an arrangement is shown. The wires 32 and 34 selectively intersect at normally open contacts 38. Each of said horizontal wires 32 receives phase shifted pulse inputs 32a-32e from a ring counter (not shown) for sequentially activating the wires in a known manner.

Actuators or keys 33 close the contacts 38 for producing coded outputs along an input/output (I/O) bus 30. Although more or less lines may be used depending on the number of keys and game parameters, in the preferred embodiment, the I/O bus has nine lines 30a-30i. If, for example, key 33C is actuated, contacts 38C are closed and outputs c and e produce pulsed outputs as high signals 32c and 32e while the ring counter is disabled and all other outputs are low. A coded output unique to the closure of said switch 38c is thus produced. An input/output couples the I/O bus 30 of the keyboard 24 to display 22 via decoder 21. In a preferred embodiment, cable 40 also couples bus 30 to the player station interface 28P, which couples outputs of the keyboard 24 to the master station 12 over the channel 16 and vice versa. It should be understood that a cordless arrangement between player stations 14 and Central Station 12 is possible.

The player station interface 28P is hereinafter described. The I/O bus 30 of the keyboard 24 is coupled via cable 40 to a buffer 42 comprising a plurality of dedicated hex-buffer gates 42a-42i respectively coupled to the lines 30a-30i of the I/O bus 30. The buffer 42 steps down signals from the keyboard 24 to an appropriate voltage for the next stage. Each gate 42a-42i may be a CD 4050 integrated circuit manufactured, for example, by RCA. The buffer 42 is coupled as shown over output lines a-i to an erasable programmable read-only memory (EPROM) 44 which acts as a decoder. The EPROM 44 may be a 2708 integrated circuit manufactured by Intel. The EPROM 44 decodes the signals over the lines a-i therefrom, and produces a coded output over its output lines a-h to a peripheral interface adaptor (PIA) 48, such as 6821 large scale integrated circuit interface manufactured by Motorola. PIA's are known as devices which provide parallel interfacing between some external device according to instructions from a central processing unit. Because the buffer 42 merely steps down the signals from the keyboard, the inputs and outputs are logically the same. The PIA 48 operates as an input/output gating device to the CPU 26 as shown in FIGS. 2 and 5, hereinafter described.

In a preferred embodiment, upon the occurrence of a key stroke, a selected output (g) of the buffer 42 goes low and provides a tag bit for setting a one-shot multivibrator or pulse stretcher 46, which is coupled to a trigger input (i) of the PIA 48. The pulse stretcher 46, including an exemplary gate, diode and RC network shown, maintains the PIA 48 in a receive mode for the inputs a-h of EPROM 44 and causes PIA 48 to produce an interrupt to the CPU 26 as hereinafter described.

5
6

Thus, when a key 32 on the keyboard 24 is actuated, selected outputs a–i of the I/O bus 30 are activated, stepped down by the corresponding buffer 42 and decoded by EPROM 44 as inputs to PIA 48. The pulse stretcher 46 produces a gating pulse to PIA 48 which responds by producing an interrupt signal. The pulse stretcher 46 holds the PIA 48 in an interrupt mode for a time sufficient to blank random noise and allow the coded inputs a–h from EPROM 44 to be received by the CPU 26. The CPU 26 recognizes a low going edge of an interrupt. Therefore, unless data on EPROM 44 is accomplished by a key stroke, such data will not be generated on the low going edge of the (g) output of EPROM 44, which is coupled to input (i) of PIA 48 via the pulse stretcher 46. Thus, if (g) goes low, another signal from the particular player station cannot generate an interrupt until the pulse shutter times out. Thereafter, data gated by the PIA 48 is transmitted to CPU over data bus 30D.

As hereinbefore mentioned, the PIA 48 is an interface device providing parallel data to CPU 26. Data output from the CPU may be gated to the player station 14 by means of other circuits in the player station interface 28P hereinafter described. Such data includes wealth information, the player's personal account or stake in the game, the bet required to stay in the game, etc.

It should also be understood that when CPU outputs data, an input interrupt will occur. During such time, the CPU recognizes and processes the input data from the station receiving output, but discards the data so obtained from that station upon completion of the output sequence. The falling edge of the signal generated by the pulse stretcher 46 generates the interrupt. However, during output the interrupt is masked. The system does not recognize the interrupt generated by the station while receiving the output.

The PIA 48 has outputs a–f which are coupled to a buffer 50 having step-up gates 50a–50f, such as 7407 integrated circuits manufactured by National Semiconductor. The buffer 50 has outputs a–f coupled to a decoder 52 (FIG. 3B). The decoder 52 includes three decoders 52a–52c such as CD4028 integrated circuits manufactured by RCA and sometimes referred to as "one of eight" decoders. The decoders 52a–52c receive selected outputs a–f of the buffer 50. For example, each decoder 52a–52c receives outputs a–c of the buffer 50 at its corresponding input a–c. Further, each decoder 52a–52c respectively receives one each of the remaining outputs d–f of the buffer 50 at a respective corresponding input d', e' and f'. Therefore, the outputs a–c of buffer 50 provide coded data, and the outputs d–f, when energized, select one of the decoders 52a–52c to receive such data. For example, when output f of the buffer 50 is high or on, outputs d and e are low. Thus, only decoder 52c receives an input f' enabling it to receive the data from the outputs a–c. Likewise, when output e of the buffer 50 is high, the d and f outputs are low, and only decoder 52b receives an input e' to render it active.

The decoder 52 is coupled to a switching device 54, which preferably includes analog switches 54a–54e such as CD4016 analog switch devices 54a–54e manufactured by RCA. Each switch 54a–54e closes or short circuits selected outputs a–h thereof in response to coded inputs from the decoder 52. In the drawing, it can be seen that the decoder 52a has outputs a–d coupled to corresponding inputs a–d of the analog switch 54a. The remaining outputs e–h of decoder 52a are coupled to inputs a–d of analog switch 54b. Likewise, decoder 52b

has half of its inputs a–d coupled to the analog switch 54d. Finally, decoder 52c has four outputs coupled to the analog switch 54e. Other outputs of the decoder 52c (not shown) are spares and not used in this particular circuit. It should be understood that while an output to display 22 is occuring, the system software does not permit or recognize an input from the particular player station. Thus, a conflict of signals is avoided.

Outputs from the decoders 52 actuate inputs to the various analog switch devices 54a–54e for selectively closing or short circuiting selected outputs thereof. For example, the decoder 52a, when energized by actuation of its d' input as hereinbefore described, transmits the coded data from the inputs a–c for driving selected ones of its outputs a–h to an "on" condition. It should be noted that alternate outputs a, c, e and g of the switch 54a are coupled to the I/O bus 30 of the keyboard 24 as leads a, b, c and d over wire 40. Outputs b, d, f, and h are joined together along a common lead to the I/O bus 30 as lead e. Likewise, the switch 54b is similarly arranged so that leads a, c, e and g are coupled in parallel as leads a, b, c and d of I/O bus 30. Common leads b, d, f, and h are coupled to lead f of I/O bus 30. Switches 54c–e are likewise coupled in parallel with I/O bus 30, but with respective common leads g, h and i coupled to I/O bus 30 as shown. If the output c of the decoder 52a is driven high, outputs e–f of the analog switch 54a become short circuited by internal circuitry thereof. Thus, when the outputs e–f of switch 54a are closed, wires c and e of the I/O bus 30 are shorted. This is analogous to the closure of a switch 33C at intersection 38C'. When corresponding horizontal and vertical wires 32 and 34 of keyboard 24 are closed at 38C' by switch 54a, it is as if switch 33C had been manually closed. The player station 14 therefore responds by providing a digital output to decoder 21 driving display 22, as hereinbefore described. In 54a–54e have corresponding outputs which are coupled in parallel with the respective normally open contacts 38 of the matrix hereinbefore described.

The player station interface 28P thus provides input data to the central station 12 from the keyboard 24, which input data is simultaneously decoded at 21 and displayed on the player statio display 22 by virtue of the closure of the selected normally open contacts 38. Likewise, the player station interface 28P couples data transmitted from the central station 12 to the corresponding player station 14 by closing selected switches in parallel with the normally open contacts 38 of the keyboard 24 for decoding at 21 and display on the player station display 22.

The PIA 48, hereinbefore described, performs other functions as well as the routing of input and output data between the player station 14 and the central station 12. The operation and programming of the 6821 is explained in detail in the 6821 manual. In a preferred embodiment, port A is programmed as an output port. Port B is an input port that has been conditioned to accept an interrupt as hereinbefore defined. Data port D accepts and transmits Data, and control port C accepts or transmits control functions by interrupt IR, read/write R/W, Reset R, Clock C, Enable EN, and station address or selection data SO, SI and SA.

The PIA 48 gates data from the CPU 26 over data bus 30D only when it is properly addressed. This occurs when a station address (SA) lead is actuated at control port C. Similarly, PIA is operative for communicating data at data port D to and from the CPU 26 over data bus 30D when conditioned by the CPU 26. Selection of

7

8

such an input or output mode of the PIA 48 is accomplished by selecting or addressing the on condition of respective input SI or output SO register selects of PIA 48. Similarly, the CPU is conditioned to read or write data only if the read/write R/W input of the PIA is properly conditioned.

In connection with the foregoing, the present invention utilizes a memory mapped system. Upon initialization of the system, codes generated in CPU 26 produce coded inputs to each PIA 48 (1–8) unique thereto. The codes condition the PIAs 48 (1–8) such that selected terminals act as inputs or outputs etc. This system software handles input output functions of the PIAs 48 (1–8). In other words, the PIAs 48 (1–8) are programmed on initialization to act in the manner desired (for example addressing input and output register selects SI and SO), such that, it is only necessary to call on a particular PIA and its output port A or input port B acts accordingly. Other systems are possible for selecting input and output function, and the like. However, the memory mapped software of the present invention has been found to be a preferred and efficient system for accomplishing the task.

When the particular player station 14-1 has been addressed by Station Address (SA) and R/W is in write or low, PIA 48 transmits data from the CPU 26 to the display 22 over the buffer 50, decoder 52, and switch 54 as hereinbefore described. When PIA 48 is addressed in a read mode, the R/W input is in a state opposite from above. Input data produced as a result of closure of certain ones of the normally open contacts 38 (resulting from mechanical key strokes) is transmitted to the CPU 26 by PIA via buffer 42 and EPROM 44 as hereinbefore described.

PIA 48 has a clock input (C), which is produced by a clock (hereinafter described) at the central station 12. The clock produces pulses which hold the PIA 48 in synchronism with all the other player stations 14 (1–8) and the central station 12.

The PIA 48 has an interrupt output IR coupled to the central station 12 over the channel 16. The IR output is actuated or goes low whenever the tag bit (t) produced by EPROM 44 drives the pulse stretcher 46 on, thereby holding input i of the PIA 48 on as hereinbefore described. The interrupt IR communicates a pulse to the central station 12 indicating that data is available from the keyboard 24 for interpretation by the central station 12.

The PIA 48 may be reset to an initial condition by means of the reset input R as shown. When the system is initially turned on, a reset pulse is coupled to the PIAs 48 (1–8) for enabling the circuits and registers of the PIAs to receive the coded signals from the CPU 26 whereby the portsof PIAs 48 are mapped or conditioned to act as inputs and outputs.

Master Station 12

For a description of the master or central station 12 reference is directed to FIGS. 2, 4A–4C and 5A–5B. The master station 12 includes the CPU 26 and general interface 28G. Each player station interface 28P is coupled in parallel with general interface 28G over its respective channel 16 including data bus 30D Each player station 14 (1–8) is operative for communicating with the master station 12 to the exclusion of all the other player stations 14 by means of an interrupt function of the CPU, which processes one interrupt at a time. CPU 26 recognizes the station by selective actuation of the re-

spective Station Address (SA) for the particular player station 14 and testing the polled station for the presence of a valid address code.

In a preferred embodiment, the CPU 26 processes data and produces outputs to the player stations 14 (1–8), the master displays 20 and indicator lights 8 (1–8). In the event of a key stroke produced at any play station, the CPU 26 completes the program instruction (i.e. line) at hand and recognizes the interrupt The CPU 26 polls the stations 14 one by one and takes in data from the interrupting station. Thereafter the CPU 26 resumes the program function. The data received from the player station is later processed in accordance with the system software.

The general interface 28G includes inverting bidirectional driver 62 including two DM 8835 integrated circuits 62W, 62R manufactured by National Semiconductor. The drive 62 is coupled into the CPU 26 data bus 30D.

Communication between the CPU 26 and player stations 14 (1–8) is accomplished by means of selectively addressing each of the player stations 14 (1–8) separately over select address line SA (1–8). Address bus 30A is coupled to the general interface 28G as shown. The address bus 30A is coupled to the channels 16 (1–8) carrying respective select address lines SA (1–8) dedicated to respective player stations 14 (1–8).

The general interface 28G includes a random access memory (RAM) 64, addressed as shown by address lines A0-A8. RAM 64 includes three RAM devices 64A-64C (shown in FIG. 4 and sometimes referred to as chips) or integrated circuits such as 6810 devices manufactured by Motorola. The RAM 64 is capable of holding at least 384 bytes of eight bit data (128 in each RAM 64A-64C) information and may be used to store values to be displayed, temporary results of arithmetic routines, system control, variable accounts, game statistics, etc.

The read/write (R/W) input to each RAM 64A-64C selectively enables each to operate in either a read or write mode in correspondence with the read or write mode of the CPU 26. Thus, in accordance with instructions established in the computer program, the RAM 64 contains or stores non-conflicting input and output data for each player station 14 (1–8) and the central station 12.

The clock line C operates the RAM 64 in synchronism with all other devices in the apparatus. A valid address VA line carries a signal that verifies that the information on address bus 30A is in fact a valid address.

In the preferred embodiment, the CPU 26 respectively reads and writes information to and from the various player stations 14 (1–8). In addition, the CPU 26 provides visual indication in the central station 12 of the information common to all the players by means of the digital display 20 and the particular information relevant to a player station 14 associated with a set of indicator lights 18 as hereinbefore noted.

The RAM 64 is coupled to the CPU 26 via the address bus 30A and the data bus 30D. The RAM 64 is a read/write device, that is, information stored in the RAM is readily accessible by the CPU 26 acting in a read mode, and the CPU 26 can change that information at a selected address in the RAM 64 when acting in a write mode. When properly addressed on the address bus 30A, the RAM 64 produces an output on the data bus 30D which is coupled to the CPU 26. Other por-

9

tions of the system, including the indicators 18, displays 20 and player stations 14, are not responsive to data on the data bus 30D unless they have been preconditioned to be responsive thereto. In other words, if the RAM 64 has been initialized to communicate with the CPU 26, other portions of the system are simultaneously initialized not to be responsive to the RAM 64. If it is necessary to change RAM 64 in any way, the address bus 30A is selectively actuated to reach the proper address in RAM 64, and data is transmitted from the CPU 26 over the data bus 30D to the input of the RAM 64.

The CPU 26 communicates with the player stations 14 (1-8) and vice-versa. The RAM 64 and the player stations 14 (1-8) do not directly communicate with each other. When communication is open between the RAM 64 and the CPU 26, communication is closed between the player stations 14 and the CPU 26.

The CPU 26 controls the indicators 18 and displays 20 in the central station 12 by means of a peripheral interface adaptor (PIA) 66. The PIA 66 includes three peripheral interface adaptors 66A, 66B and 66C such as 6821 integrated circuits manufactured by Motorola. Each PIA 66 receives coded data from CPU 26 over data bus 30D representing information commonly available to all players in the game, e.g., Win, In/Out, Wealth Exceeded, Player Up, and Pot Value Information. A selector 88, coupled to PIA 66, enables it to operate selected outputs for actuating indicators 18 and displays 20.

The PIA s 66A-66C are coupled to the data bus 30D as shown. The PIA s 66A and 66B are selectively enabled to be responsive to the data on the data bus 30D for providing input to indicators 18. The data is communicated from the PIA s 66A and 66B to a solid state switching device 68 which includes a plurality of solid state switches 68a-68h, such as Sprague UD4181 power drive integrated circuits. The switching device 68 selectively enables certain ones of the lights: Win, 18W (1-8), Player Up 18P (1-8), Wealth Exceeded 18WE (1-8), and Player In 18M (1-8), depending on the game condition and the status of the player in question.

In a preferred embodiment, all of the outputs of the PIA's 66A-66B are in a high or activated state. Coded information from the CPU 26 causes one or more of the outputs of the PIA s 66A and 66B to become low for causing the selected switches 68a-68h to drive one or more of the indicator lights 18 on.

The PIA 66C is dedicated to be responsive to the data from the CPU for driving selected inputs of the digital display 20 to an on condition thereby creating an alpha numeric display of information relevant to the game. The PIA 66C has one set of outputs a-h coupled to booster 82, including a pair of booster circuits 82a and 82b, such as integrated circuits 7437, manufactured by National Semiconductor.

The booster 82 raises the level of the outputs a-h of the PIA 66 to an appropriate level for driving displays, hereinafter described. Outputs i, j and k of the PIA 66C are not amplified.

The booster 82 outputs f-h and PIA 66C outputs i, j and k are coupled to decoder 90. The decoder 90 includes three one of eight decoders 90a, 90b and 90c, such as 7442 integrated circuits manufactured by National Semiconductor. The one of eight decoders 90a-90c cooperate as the decoder 52 in the player stations as described above.

Outputs a-e of the booster 82 are coupled via respective pot display connectors 92a-92d to pot displays

10

93a-93d such as HP. 5082-7300 manufactured by Hewlett-Packard. The pot displays 93a-93d each include six display windows 94a-94f each of which receives and decodes the inputs a-e for producing alpha numeric displays in each of the windows 94a-94f of the displays 93a-93c.

Outputs i-k of the PIA 66C are coupled to strobe inputs i'-k' of the decoder 90. As a coded input from decoder 82 appears on the lines a-e of each window 94a-94f of the pot displays 93, the strobe inputs i'-k' cause its respective decoder 90a-90c to strobe selected outputs a-x in succession. Thus, the windows 94a-94f of each pot display 93a-93d are selectively activated with a numerical symbol representing data from the central processing unit 26.

A selector 88, such as a 74S138 one of eight decoder manufactured by National Semiconductor, on the general interface 28G has inputs a-c and an inverted VA input. The inputs a-c provide eight combinations of binary logic for controlling the selector 88. Respective outputs a-h of the selector 88 are coupled to PIA's 66 respective select address inputs SA (1-3) and SA (1-8) of the player stations 14 (1-8). When a VA signal coupled to enable selec 88 is present and outputs are available on the lines a-h of the selector 88, one or two of the PIAs 48 or PIAs 66 is selected for communication with the central station over its respective select address lines SA (1-3) or SA (1-8). Thus, means is provided for selectively utilizing the selector 88 as a decoding device for each of the player stations 14 as well as a decoding device for selectively operating the various indicators 18 and displays 20.

Address bit A5 shown in FIG. 4B is provided for assuring that the outputs a, b, and c of the selector 88 are not confused with the outputs a-h of the same selector when in communication with the player stations 14 (1-8). This occurs as follows: a tag bit provided by the address bus 30A at line A5 is coupled to enabling inputs EN of the PIA's 66A-66C. The tag bit A5 is coupled to similar enabling inputs EN on each of the PIA's 48 for the player stations 14. However, the bit A5 is inverted (See FIG. 4B) between the general interface 28G and the player stations 14 so that when A5 is present, PIAs 66A-66C are enabled and the player station PIAs 48 (1-8) are disabled, and the absent A5 is converted into an enable signal for enabling the of the various player stations 14 (1-8). Thus, the 88 operates for selecting the various PIA's 66 and 48 only when a selected enable signal is available from the CPU 26, and the use of address bit A5 differentiates between the local indicators at the general interface 28G and the remote indicators at each of the player stations 14.

In FIGS. 2 and 4A, signals from memory decoder 116 and address bus 30A are coupled to inputs of logic 70. In the preferred embodiment, a signal A15 corresponding to the addresses not allocated to the PIAs and a signal corresponding to the addresses not allocated to RAM (8000) are used. These signals are inverted by invertors 71 and 73 respectively and then OR'd by OR gate 77 to produce a signal that is in a high state when either a PIA or RAM is addressed by the CPU. The output of OR gate 77 is used as an input to enable NAND gate 79, while the other input to NAND gate 79 is the R/W signal. The output of NAND gate 79 will then be in the low state only when the CPU is in the READ mode and either a PIA or RAM is being addressed. Signals $\overline{VA}$ and $\overline{C}$ are OR'd by OR gate 83 to produce a signal which is in the low state only when the

clock and valid address lines are in the high or true state. The output of OR gate 83 and NAND gate 79 are inputs to OR gate 81. OR gate 81 therefore is in a low state only when the CPU is in a valid READ mode involving either a PIA or RAM. Otherwise, OR gate 81 is in a high state.

The output of OR gate 81 is applied to the input of 62I and the read-enable-on-low input of bidirectional devices 62R–62W. The output of gate 62I is applied to the write-enable-in-low input of 62. When the CPU is reading from a RAM or PIA, the output of OR gate 81 is low and therefore driver 62 is in the READ mode. When any other section of memory is addressed, a write operation is occurring, or an invalid memory location, the driver 62 is in a write mode. This prevents invalid data from the general interface from interferring with data on the CPU bus, i.e. cross-talk. Note that a corresponding bi-directional driver 102 operates in a similar manner under CPU control so that signal direction is maintained.

Central Processing Unit 26

The central processing unit CPU 26 is described hereinafter with respect to FIGS. 2 and 5A–5B.

CPU 26 is coupled to the general interface 28G described hereinbefore over the data bus 30D and the address bu s 30A, which is a subset of the address bus of the CPU. Data output from the CPU 26 is coupled to the data bus 30D through a bi-directional inverting driver 102 which may be an 8835 integrated circuit similar to the bi-directional driver 62 hereinbefore described. The data output of the CPU is thus inverted. Double inversion by the drivers 62 and 102 assures compatable polarity of the data signals from the CPU 26 and the general interface 28G.

The CPU includes a micro-processor 104, which may be a 6800 integrated circuit manufactured by Motorola. The micro-processor 104 communicates with the data bus 30D as shown. Similarly, the micro-processor 104 communicates over the address bus 30A via a driver 108 which may be a DM 8097 manufactured by National Semiconductor. A read only memory ROM 106 includes a plurality of ROM circuits 106a–106d, such as 2708 EPROMs manufactured by Intel. The ROM 106 is loaded with the program for operating the game in accordance with the flow charts hereinafter described. The CPU 104 addresses the ROM over the address bus 30A for accessing information relative to the game program, which information is coupled to the micro-processor 104 over the data bus 30D. A decoder 116 is responsive to certain address lines on the address bus 30A for producing outputs indicative of the particular memory segments addressed by the micro-processor 104, one example of which has been described with respect to logic 70. Outputs of the decoder 116 are utilized for logically gating other portions of the system hereinafter described. A clock 110 is coupled to the micro-processor 104 and to other portions of the system over the clock lead C as hereinbefore noted. The clock 110 produces pulses for driving the system in synchronism.

As with most computer operated systems, the computer or micro-processor 104 shares its time among various portions of the system. Accordingly, means is provided for selectively gating the micro-processor 104 so that it selectively communicates with various portions of the system without contention. Further, the peripheral devices coupled to the micro-processor 104

produce signals which are selectively received or blanked in accordance with means for sorting or keeping track of the various signals. Accordingly, selected outputs of the micro-processor 104 are logically coupled to various peripheral devices, hereinbefore described, for selectively actuating certain ones and deactuating others in accordance with the operation of the system.

The operating system of the 6800 micro-processor is described in a 1978 publication of Motorola, Inc., entitled M6800 Micro Computer System Design, Data, 2nd printing, which publication is incorporated herein by reference. The control signals and operating system of the present invention are comparable with the micro-processor described in said publication.

The micro-processor 104 is operative for communicating with the selected player stations 14 (1–8) for transmitting information to such stations. Similarly, the micro-processor 104 is conditioned for receiving information from the player stations 14 (1–8) in response to interrupts and other signals necessary for such communication. The micro-processor 104 communicates in accordance with its interpretation of the instructions stored in its ROM 106.

In FIGS. 5A–5B, various individual circuits of the CPU 26 are illustrated in detail. Micro-processor 104 has certain inputs and outputs including the interrupt IR, read write R/W, valid address VA, reset R, clock C, data lines DO–D7 and address lines A0–A15.

Interrupts IR are communicated to the micro-processor 104 by each of the player stations 14 (1–8) and as described above.

The CPU generates read/write R/W pulses for selectively enabling and disabling devices in communication with the CPU in accordance with the operating systems of the micro-processor. For example, the micro-processor 104 reads the program from ROM 106. The micro-processor 104 reads and writes to the RAM 64 in the general interface 28G by means of read write line R/W.

The clock produces clock pulses for driving the micro-processor 104 and other devices hereinbefore described in synchronism. The clock 110 may also produce other time signals as necessary. The clock 110 also produces a reset upon actuation of the system during the power up or initialization phase of the system operation. Initialization occurs in accordance with ordered instructions in software. Instruction manuals of the various IC's describe initialization requirements which need not be described here.

GAME PLAY

The actual use of the invention involves following a procedure not unlike the normal play in a game of poker. Each player, by means of the keyboard, is able to communic.ate with the central station for performing certain betting and housekeeping tasks.

Table I below lists the keys available for use on the keyboard by symbol printed thereon and by key name, When the key is actuated, the display shown on the player's station and the central station, if appropriate, is listed. The key function(s) is summarized in the right hand column. Table II lists the indicator lamps by color and the meaning of the same with respect to a particular player's station or status.

5,159,549

13

TABLE I

Key Symbol(s)	Name	Display/ Indictator	Function
C	CLEAR	Zero	Clears player station display and station input memory to Zero
.	Decimal Point	Decimal	Separates dollars and cents in display
WA	Wealth Account	$	Displays wealth of player at Player's Station only
INC/WA	Increase Wealth Account	$	$ + INC/WA increases wealth account of player
DEC/WA	Decrease Wealth Account	$	$ + DEC/WA decreases wealth account of player
PA	Personal Account	$	Displays total amount bet by player in current game
BA	Bet Account	$	Displays amount to stay in
HA	Hand Account	$	Largest personal account in hand
OS/ACK	House Acknowledge	#	# + OS/ACK by Bank or House indicates house mode for named player station
	Player Acknowledge		OS/ACK by player adds or substracts wealth attributed by house in WA above
OUT	Out	Green light goes out	Player withdraws from game
BET/RAISE	Bet/Raise	$	Bet and/or Raises Display sets new WA, allows game entry in ante phase
	Win/ Acknowledge	$	Acknowledges a win to permit transfer of pot to winning player's wealth account
	New Game	Zero	House starts new game
0-9	Numbers	$ or #	To display $ To indicate a player station #
ABORT	Abort	Zero	exit house or bank operation without transaction

Legend
$ = Numbers indicative of money or points
# = Numbers indicative of station identity.

TABLE II

INDICATOR LAMPS	PLAYER STATION STATUS IF LIT
Green	Player In
White	Player Up
Amber	Wealth Exceeded
Red	Win

Some of the keys have multiple functions, noted above, depending upon whether it is used by the individual player as a player or by the house in performing housekeeping tasks hereinafter described.

In a game of poker or other game of chance where players compete against each other using chips and the like to represent wagers, the players purchase the chips from the house or bank in various denominations, and use the chips for making wagers in one or more games by placing the chips in a pot. Normally, a game begins, if the rules so provide, by each player placing an ante or initial bet in the center of the table or pot. Thereafter, the cards are dealt, and the player to the left of the dealer has the option to check, meaning pass, or bet a specific amount of money or drop out. The first player

14

to bet places chips representing the wager in the pot. Other players wishing to remain in the game must meet the initial bet. In addition, any player or players in succession may raise the bet by adding to the bet amount an additional amount representing a raise. Players thereafter must meet the initial bet plus the aggregate of raises in order to stay in. Play continues until all of the players but one have dropped out. The remaining player is declared the winner and sweeps the pot, thereby accumulating wealth.

As hereinbefore described, the winning player normally stacks the chips in accordance with the denominations while another hand is dealt. Play may continue until all of the players leave the table or until an agreed time. If a player decides to drop out of the game, he may cash the chips by returning the same to the house in exchange for the equivalent value in money. At the end of play, the chips are all cashed and stacked and returned to a receptacle for use at another time.

In the present invention, the game of poker is played in essentially the same way as hereinbefore described. Players ante to enter the game, receive cards, place and raise bets, drop out and ultimately a winner is declared. The difference is that, with the present invention, no chips change hands because the device tallies and keeps track of the amounts represented in each player's account and the pot in accordance with the normal rules of poker.

Banking Phase

In order to initiate the first game, the system is turned on. At this state, the object is to distribute wealth to the players in a way similar to the distribution of chips. When the system is turned on, one station, for example, player station 14-1 is automatically designated as the house. One at a time, the players deposit funds with the bank or house. Thereafter, the person operating the bank or house player station 14-1 presses the player station number (#) depositing money and OS/ACK. This conditions the particular player station, e.g., 14-2, to receive a credit for the amount deposited. The house hits the CLEAR button, the amount deposited, e.g. $1,000.00, and then hits the INCR/WA button to transfer the funds to the account of the player station in question. The amount then appears on the display of the player station receiving the wealth. That player station player hits his OS/ACK key to acknowledge that the amount is correct and received. If the player thereafter hits his WA button for wealth account, the $1,000 should display on his individual display only. The aforegoing series of operations is repeated for each player entering the game.

Ante Phase

The next stage of play is the actual beginning of the game. Games normally begin with the ante phase. All eligible players enter the game at this time. The players may enter in any order because sequence of play is not enforced at this time. The entry of the first ante bet begins the game. For example, player station 14-2 hits 10 and the RAISE/BET key. $10 appears on the pot display of the central station, player station 14-2 IN/-OUT green light turns on and any remaining wealth at player station 14-2 appears on his individual display. In the example above, if player two had $1,000 in the original wealth account, $990 would appear, representing the original wealth amount less the $10 ante.

15

All players who accept the initial ante now become part of the game in progress. Such players may accept the ante by merely hitting their respective RAISE/-BET key which causes the pot amount to increase $10 as each player enters the game and the green light for the particular player to go on. Each player receives an indication of his or her remaining wealth and players may drop out by pressing the OUT button.

Raise/Bet Phase

After the cards are dealt, the first active player who makes a bet starts the Raise phase of the game. For example, player station **14-4** may open with a $10 bet. The $10 is added to the previous amount in the pot display and the remaining wealth is displayed on the display of the player making the bet. As in the ante phase above, any player may leave the game at any time by pressing the OUT buttom. Once this occurs, the player may not re-enter that particular game. This is true for any game phase.

After the first player bets, the Player Up or white light appears on the station for the next eligible player to the left. If, for example, the player at station **14-4** began the game or opened with a bet and the player at station **14-5** had previously droppdd out, the next eligible station player would be the player at **14-6**. The Player Up light at **14-6** would therefore go on. Player six may call the bet by merely pressing the RAISE/-BET key, or he may raise the bet by hitting numbers indicating the amount of the raise and the RAISE/BFT key. (For example, 2 and 0 for $20 and the RAISE/-BET key). The original $10 bet plus the $20 raise will be added to the pot display. In the example above, the bet is now $30 to the next eligible player. This amount will enter in his display along with the energization of the white light. Assume that there are only three players in this particular game, e.g. 1, 4 and 6, player one must meet the initial $10 bet plus the $20 raise in order to stay in the game. Thus, $30 appears at his display when his white Player Up light goes on. When it is the 4th player's turn, because player four had made the initial bet, he need only to match the $20 raise. Therefore, $20 appears in his display along with the white light indicating that it is his turn to either call or raise the bet or go out. Calling or raising the bet activates the next eligible player station.

Win Phase

The betting pattern continues as the game is played with the cards until the winner is declared. In an actual game of poker, if all the bets are called, according to the rules, the player making the last bet must show his cards to the other players. If the cards are winners, the other players hit their respective OUT buttons. As a result, the red WIN light goes on at the station of the called player who had not dropped out. If another player shows better cards, the called player and other players hit their respective OUT buttons and the WIN light lights at the player showing the better cards. The final pot for winning amount is displayed in the pot display and in the winner's station. The winner hits the RAISE/BET key to acknowledge and accept winnings as indicated on the pot of the central display. This amount is added to his wealth amount, which is displayed to him.

It should be understood that multiple winners may be declared (i.e., a shared pot). For example, the sequence may be initiated by a decimal numeral key stroke indica-

16

tive of the pot percentage claimed as won preceding the OUT key stroke. When all players are either out or claiming to be winners and the values claimed equal one hundred percent of the pot displayed, the winnings are displayed in the respective winner's display and each acknowledges the amount won.

New Game Phase

A new game is begun when the player representing the house hits the RAISE/BET key. At that time, all of the indicator lights are turned off, the pot display is cleared, and everyone's remaining wealth is displayed in their individual respective displays. The first player thereafter making an ante bet starts the betting process again.

Cashing Out

Any player may cash out by requesting the same from the house. The house presses the player station number, e.g., 2, plus the OS/ACK key. The house hits the CLEAR button and the amount withdrawn, e.g. $1595 and the DEC/WA (Decrease Wealth Account) button. The player examines this figure and, if correct, he hits the OS/ACK to acknowledge that the transaction is correct. His remaining wealth appears on his display. If the decremented amount equals the wealth account, $0 is displayed. Thus, the player is effectively out of the game and cannot bet unless and until the wealth account is replenished

Wealth Exceeded

In the preferred embodiment, any time a player exceeds his wealth amount by making a bet which is more than the amount in his wealth account, at that time the amber WEALTH EXCEEDED light for the player goes on and the player is precluded from making a bet. The player may increase his wealth amount by paying in as described above, in the banking phase, after which the player may make a bet.

It should be understood that the present invention may be used as a tallying device in any game in which players compete against each other or the house, as in Black Jack. However, a different program must be provided to accomplish such result. The present invention is most conveniently and preferably applied to the game of poker in various forms as hereinbefore described.

GAME LOGIC AND FLOWCHARTS

In FIGS. 7A-7B and 8A-8C, there are shown two flow charts of the system. In FIG. 6, GAME SEQUENCING is shown. After start up, the system is designated to move through a series of game phases in an ordered sequence In the ANTE PHASE, random entry into the game is allowed. Ante bets are processed between the Taskhandler and Ante software in primary loop I. If any player in the game initiates a raise over and above the initial ante, the GAME SEQUENCING moves into GAME software. Thereafter, bets and raises are strictly ordered and random entry is forbidden. Thereafter, system software moves between game functions and the Taskhandler functions in Loop II. In the preferred embodiment, after all bets and raises have been made and all but one player has been eliminated, the GAME SEQUENCING goes into the WIN software. Win acknowledgement tasks associated with the win phase of the game are processed in Loop III. After all wins are acknowledged, GAME SEQUENCING

17

18

moves to the NEW GAME SOFTWARE upon actuation of the RAISE/BET (new game) key by the house. Tasks are processed in Loop IV. Once all new game tasks are accomplished (e.g. calculations and initializations are complete), GAME SEQUENCING goes back to ANTE as shown. The system thus controls the instruction sets available for each phase of the game. In FIG. 7A–7B a more detailed general system flow chart is shown. Operations are written in rectangular boxes and questions or inquiries are written in diamond-shaped boxes in accordance with known flow chart drafting techniques.

During the start up of the game, individual players pay in and increment their wealth accounts in accordance with the previously described sequences. Thereafter, players enter a game playing sequence. The sequence includes the ANTE phase, a GAME (raise/bet) phase, a WIN phase and a NEW GAME phase as hereinbefore described.

The system software as outlined in the flow charts of FIGS. 6–8 anticipates the various phases. Power On at 200 indicates that the system has been initially turned on. The Initialization operation at 202 results from a reset pulse from the CPU 26 for initializing the various memory devices and the lik to an initial condition. Further, memory displays and the like are initialized to begin the game e.g., the game status lamp registers are cleared in memory and then the various game status lights are turned off, indicating no activity.

The system goes to Instructed Return Point at 203 after initialization at 202. Because the system cycles through the various loops I–IV, it has instructions in software for cycling the instructions which, in effect, skip earlier instructions which are not needed. Instructed Return Point 203 is a flow chart routing mechanism for instructions which will be further discussed hereinafter.

Start task function 204 begins a sequence of tasks, i.e., routing various program sequences to sub-routines and the like. The system begins in ANTE phase. See FIG. 6. A Task Present inquiry at 206 asks the system whether a task has been initiated. If the response is NO, as indicated by N, the system loops back to the Start Tasks routine at 204. If the answer is YES, as indicated by the Y, the system proceeds to a Determine Source routine at 208. The question is then asked whether the task is a Clear task at 210 or something else. A Clear task means that the system shall operate the Clear Source operation at 212 through a Clear Status Bit function at 214 and return to the Start Task function at 204.

The Clear Status Bit function 214 is a housekeeping and programming task which is known in the art. Although not always noted, the Clear Status Bit function 214 is shown in the drawings at various places, and it should be understood that it occurs before each cycle.

If the Clear inquiry 210 is a negative, the question is then presented whether the function or task is a Bank Task at 216. If the answer is affirmative, a Test for Bank Mode 217 and a Test for Function at 218 is made for the function. Such functions include Reassign the house or banking station at 220, Increase/Decrease wealth account WA at 222, and an Abort at 224. The affirmative of Reassign inquiry is coupled to Reassign Routine at 226. After completion, the system returns to the Start Tasks at 204 through Exit Bank Mode 231 and Clear Status Bit at 214. Similarly, Increase/Decrease WA Routine at 228 operates in response to an affirmative inquiry from the Inc/Dec WA Inquiry at 222. Finally, if

an error is made in the sequencing of keys, the operator may hit the ABORT key which enables an affirmative of Abort Inquiry at 224 to operate Abort Routine 230 and return to Start Tasks 204 via the Exit Bank Mode 231, Clear Status Bit 214 and Instructed Return Point 203. Further, a negative response to Abort Inquiry 224 at this stage of play causes the system to Abort also. This appears redundant. However, this software sequence avoids a potential program loop by default.

If the Bank Task inquiry at 216 is negative, the system inquires if Decimal String=0 at 232. The Decimal String is a representation of the series of numbers which precede the operation of a function key. If numbers do not precede the function key code, the answer is affirmative. For example, if a player wishes to make a bet of $10, the player activates the 1 and 0 keys and then the RAISE/BET key. The Decimal String is not equal to zero in this case. If, however, a player wishes to meet a bet, but not raise it, the player merely activates the RAISE/BET key. In such case, Decimal String is equal to zero. By default, the system automatically credits the pot in the amount of the unstated bet. The Decimal String is a way of determining whether the particular task is purely a betting task or some other player task.

If the Decimal String=0 Inquiry 232 is affirmative, it indicates that a bet or ante has been met; a bank function is occurring; or a player function is occurring. Out inquiry at 234 following Decimal String=0 inquiry asks whether the player is in or out. If the Player Out inquiry at 234 is affirmative, a Remove Player Routine 236 is employed. Thereafter, a question is asked at 238 whether there is One Player Left. If the response is negative, the system goes to Clear Status Bit 214 and returns to Start Tasks at 204. If the response is affirmative, Win Routine 240 is activated. The red light at the winner's station is activated and the pot amount is displayed in the central display and at the particular player station as well as the central display. As hereinafter described, the system deals with a win situation by interaction with the bank and the particular player described below.

If the Out? question at 240 is negative, i.e., if the player is not taking himself out of the game, then the function following the Decimal String=0 inquiry 232 may be a bet. The Bet Key inquiry at 242 may be affirmative or negative. If negative, the system goes to Player Station Function Routine 244. Such a player station function may be to acknowledge an increase or decrease in the wealth account or re-assignment from the house or bank. The particular inquiries are not shown in the drawing because it is believed a verbal explanation is sufficient and the drawing may be unduly cluttered thereby. If the Bet inquiry at 242 is affirmative, the system moves to go to proper routine (as determined by instructed return point) at 246. In this way the software returns to the phase of the game it has been instructed to be in at such time, i.e., Ante Routine 248, Game (Bet Raise) Routine 250, Win Routine 252 and New Game Routine 254. If the system is in any of the foregoing routines, it may loop back through the Task-handler via start tasks at 204. Once the system moves from the Ante Routine 248 to the Game Routine 250, it does not return until a new game starts. Similarly, if the system moves to the Win Routine 252, the loop back through Start Tasks 204 bypasses the Ante and Game Routines. Finally, in the New Game Routine 254, the system does the new game functions and returns to Start Tasks 204, by-passing the routines associated with win

acknowledge, ante and raise bets. Each of the foregoing routines 248–254 is described hereinafter with respect to FIGS. 8A–8C. At present,however, further inquiries are required before this flow chart is satisfied.

If the Decimal String=0 inquiry at 232 is affirmative, it is apparent from the foregoing that a player is either performing certain player station functions or making a bet without a raise. If the Decimal String=0 inquiry 232 is negative, it is possible that certain house or banking functions are in process or a raise situation has occurred. For example, during a re-assignment, the house depresses a player station number and then a function key. Similarly, during an increase or decrease wealth account function, the house depresses numbers plus the function key for increasing or decreasing numbers plus the function key for increasing or decreasing the wealth account. Thus, the Decimal String preceding the function is not equal to zero. (Also, in a raise bet situation, a player raises a bet by first placing the amount of the wager or raise on the keyboard and then hitting the RAISE/BET key.)

In the situation where the Decimal String inquiry at 232 is negative, a Bank Key inquiry is made at 256. If affirmative, the system switches to Bank Mode Routine at 258. (A negative response to the Bank Key inquiry 256 indicates a betting situation). The system may not be in a betting mode and a bank mode at the same time. Thus, there is a check on the house mode to prevent cheating.

If it is a bank function, Bank Mode Routine 258 is executed and the system returns to the Start Tasks Routine at 204. In operating through such a Start Tasks Routine at 204, it can be seen from the flow chart that the system will run through Task present at 206, Determine Source at 208, Clear at 210, look at the Bank Task inquiry at 216, and Test for Bank Mode 217, and Test for Function at 218. The subsequent inquiries at 220 and 222 as well as the ABORT at 224 and Exit Bank Mode at 231 are made as hereinbefore described.

If the Bank Key inquiry 256 is negative, then a non-zero Decimal String is a Raise Bet situation. A Raise/-Bet inquiry is made at 260. If the answer is affirmative, a flag is set at Set First Bet Flag 262, and an Ante inquiry at 268 is made as to whether the Raise Bet is an Ante. If the answer is affirmative, return via Ante Routine 270 is operative to hold the system in Ante Routine at 248. If the answer is negative, the system moves to Go To Game Routine at 272 and then Go To Proper Routine at 246 as shown. Finally, if the Raise/Bet inquiry at 260 is negative, there must be an error and a system Error Routine 274 is operative to loop the system back to the Start Tasks at 204.

Select Game Phase

Referring now to FIGS. 8A–8C (see FIG. 8D for Drawing Arrangement), the flow chart resumes with Return Via Ante 270, Instructed Return Point 203, Clear Status Bit 214, and Start Tasks 204 illustrated in FIGS. 7A and 7B. As hereinbefore noted, once instructed to return to a particular game phase routine, the system stays in that routine until instructed to move on to another.

The above concept is illustrated in FIGS. 8A–8C as follows. In Ante Phase it is assumed that play is about to begin in such a way that players may ante in random order As hereinbefore stated the system interprets an initial ante as a first raise, because the pot starts at zero. Thus, a non-zero decimal string preceeding a player

RAISE/BET key stroke is interpreted as a Raise. The question Raise ? at 280 initiates an inquiry or test to determine which of two possible returns is possible. For example, if the answer to the Raise inquiry at 280 is affirmative, such raise may be the first raise of the game e.g. the initial ante. It may, however, be a second raise, that is, an increase in the initial ante, not just a matching thereof. For example, if a player wishes to meet an ante, the player strikes the RAISE/BET only after the initial ante has been entered. Thus, the decimal string preceding the key stroke is zero and the system interprets the key stroke as a BET. In usual play, if after the players have all entered an ante, a player wishes to make an additional bet i.e. a raise, the player depresses the bet amount plus the RAISE/BET key. The system interprets this as a raise. If the ante has occured, this raise, occurring after the ante, is interpreted at inquiry 282 as a secon raise whereby the system exits or goes to game routine at 246G.

In the Game Phase, betting order is important. Therefore, once a bet is made, and a raise of such bet has been made, the order and sequencing is fixed. The system will not go to the game routine before ante bets are complete. Therefore, a negative response to the Second Raise inquiry at 280 means that the system is entering the ante phase, and the system produces a command to raise an Ante Flag at 284. In the ante phase a non-numerical raise/bet keystroke is merely meeting a bet or meeting the ante.

If the ante flag at 284 goes up, the system executes Set Initial Ante at 286 and moves to process the ante by inquiries whether the bet is less than or equal to the wealth account at 283. A negative response produces a Wealth Exceeded Error at 285 and a loop to return via ante 270 as shown. If the response to inquiry at 283 is affirmative, the player has sufficient funds or points to stay in and the player in status lamp at 287 is turned on. The player's wealth account is decreased at 289 and the pot is increased at 291. Thereafter, the system returns via ante 270.

After the initial ante, a negative response to raise inquiry at 280 results in a Has Ante Occurred? inquiry at 293. A positive response to the inquiry causes the system to execute Set Bet Equal to Ante at 295. Thereafter, processing proceeds as described above via the inquiry at 293. A negative response to Has Ante Occurred 293 produces an error at Cannot Enter Without Ante routine 297, because a player cannot ante nothing or zero in order to play, (i.e. first raise has not yet occurred).

The system continues to loop back to Return Via Ante 270 as long as the second raise has not occurred. Once it does occur, the system exits the Second Raise inquiry at 282, moves to Go to Game Routine 246G, return via Game Routine 247, and Clear Status bit 214 to Start-Tasks in Game Mode 204G via the Game Routine (i.e. Loop II in FIG. 6).

As mentioned above, after a second raise has been made player sequencing is important. Thus, an Is Correct Player Up inquiry is made at 292 after Start Tasks 204G. If the response is negative, an Out of Turn Error 294 occurs and the game sequences back to the Start Tasks 204G via the above noted loop. If the response to Is Correct Player Up at 292 is positive, then the bet is processed at Calculate Bet Routine 296. An Is Bet Greater Than Wealth Account Inquiry is made at 298 as to whether the player has exceeded his wealth. If the answer is affirmative, the Wealth Exceeded Error occurs at 300, sending the system to Start Task at 204G via

Go To Game Routine 246G. If the response is negative, the Compute Accounts Routine at 302 (FIG. 8B) provides information as to the wealth amount, the bet amount, the pot amount, and other parameters.

For purposes of explanation, various error sequences are noted (e.g. Out of Turn Error 294 and Wealth Exceeded Error 300). However, the system software uses essentially the same error routine whenever an error occurs. That is, the system cycles back to the beginning of the loop where the error occurred and the player in error receives a display of all eights (888888) on his player station display.

After a bet is completed, a Player Status Routine 304 moves the player out of up status and increments to the next available player at Increment Game Station Number (GSN) 306. An inquiry is thereafter made at 308 as to whether the new GSN equals 9. A positive response engages Set New GSN=1 at 310. The logic is that because there are only eight player stations, if player 8 is the last one to make a bet, then the game must be moved up to the next player station, i.e., one. If the New Game Station Number GSN is not 9, the logic moves to the next inquiry as to whether the new GSN=Old GSN at 312. A negative response means that there is a player in the game available to make a bet. Therefore, the software executes to Set Next Player Up at 314 and Calculate and Display the Bet Account which displays the amount to stay in game on the player station and turns on his Player Up lamp, for the particular player and returns to Start Task at 204G, at which time the player may call the bet, raise the bet, or drop out in accordance with the game.

It should be understood that the system software can bypass a player station not in the game for functions, etc., but the system polls the stations in order. Then a station that is out is still counted, the GSN increments and the system moves on.

If the response to the question of whether the new GSN=Old GSN at 312 is positive, the system responds by executing Only One Player, Indicate Win+Display Amount 318. At such time the red light on the winning player station is activated and a display amount of the pot is transferred or displayed simultaneously in the pot display and in the player station display. The software then moves to Go To Win Routine at 346W, Return via Win Mode 322, Clear Status Bit 214, and the system moves on to the Start Task in the Win mode at 204W.

In the Win mode, the system inquires whether a Win Acknowledgement is Correct at 324 (FIG. 8C); that is, has the correct winning player pressed the Win Acknowledge button. If incorrect, Win Acknowledgement Error Routine 326 (similar to Errors noted Above) is executed, whereupon the system cycles back to the Start Task in the Win mode at 204W. If the proper player acknowledges the win by an affirmative at 324, the system goes to Compute New Wealth and Display Routine 328

The system moves then to the Go To New Game Routine at 330, return via New Game 331, Clear Status Bit 214, and the Start Tasks in the New Game Mode at 204N. A Bank Start New Game Inquiry is made at 334. A negative response means that the wrong player has pressed the RAISE/BET key for starting a new game and New Game Error Routine at 336 returns the system to the Start Tasks at 204 N. An affirmative response from the Bank Start New Game inquiry at 334 causes the system to Re-initialize and Clear Flags at 338 and Go To Ante Routine at 246A. Thereafter, the system

displays each player's respective wealth account at the respective display for each player by means of Display Wealth Account Routine 340, Clear Status Bit 214, and the system recycles to the original Start Tasks at 204, shown in FIG. 7A.

There are other routines for operating various logic sequences in the game which are not described in these flow charts. However, it is believed that because certain routines such as calculating, adding, subtracting, multiplying and dividing are readily known by those skilled in the art, such a description herein is believed to be unnecessary.

Further, multiple winners may be accounted for by a non-zero decimal string preceding an OUT key stroke. Thus, the player is counted out but may be later counted as a winner requiring acknowledgement in the win phase.

It should also be understood that the system may provide other types of game play. For example, a player may purchase a wealth account at the rack track and place bets at a remote location from the betting window, such as his restaurant table. The system would require identifying the player station and player game entry device such as a credit card or card entry device.

What is claimed is:

1. An electronic data processing and display system for use in a game wherein a plurality of players wager in turn on the overall value of their respective holdings relative to the other players' holdings of combinations of tangible indicia acquired in turn from a preset array of indicia in a random sequence, each indicia uniquely representing a designed value and different combinations thereof denoting order of entitlement to win the game and the accumulated wagers of all players for that game, said system comprising

plural player station means respectively associated with each player for entry of data representative of wealth for purposes of the game and amount of wager by the respective player at applicable points in the game, each of said player station means including station display means,

central data processing means responsive to wager and wealth data entries by the respective players at applicable points in the game for computation and furnishing of data to all of the player station means, and to central display means operatively associated with said central data processing means, said furnished data being representative of the accumulated wagers of all players to that point in the game and the amount remaining to be wagered by the player associated with that player station means to remain in the game on that player's next turn for respective display on the central display means and each of the station display means, said furnished data further being representative of wealth of each player at that point in the game,

means responsive to the furnished data representative of wealth of any player for limiting the display thereof to only that player station means associated with the player whose wealth is represented by that data and

selecting means operatively associated with said central data processing means and with each of said player station means for selectively designating one of said player station means as a banking entity responsive to entry of initial wealth data from each of the player station means and to wager data and added wealth data entered at applicable points in

23

the game to account for changes in each player's wealth data as the game progresses.

2. The system of claim 1, further including interrupt means operatively associated with said central data processing means for inhibiting said central dat aprocessing means from responding to data entries from any player station means other than a preempting player station means from which data entry has been commenced without competition from any of the other player station means, and for so long as such data entry continues from the preempting player station means.

3. The system of claim 1, wherein said central data processing means includes sequencing means for selectively activating each of said station display means to indicate the player whose turn it is to wager.

4. The system of claim 1, wherein each of said player station means further includes means for entry of data indicative of withdrawal of the associated player form the game, and wherein said central data processing means is further responsive to data entered from the player station means indicative of withdrawal of all players except one to add the data representing the accumulated wagers of all players up to that point in the game to the data representing the wealth of the remaining player.

5. The system of claim 4, wherein the furnished data representative of wealth of a player is furnished by the central data processing means to the player station means associated with that player to the exclusion of all others for display on the respective station display means upon entry of data from the last-names player station means indicating demand for such wealth data.

6. A dedicated electronic data processing system for a game in whic players successively wager against each other on hands of playing cards held by each player until a player is entitled to the cumulative amount wagered by virtue of holding the winning hand, said system comprising: plural sets of data entry and display stations, one set per player, for entry of data representing player purchased allocations for wagering and representing player decisions to remain in or withdraw from the game at applicable points in the game; a central

24

data processing and display unit for selectively processing data entered from the individual stations on a non-interfering basis to perform at any point in time calculations including the then-current cumulative amount wagered, the then-current minimum amount each player must respectively wager to remain in the game, and the then-current purchased allocation remaining for each player and whether or not the amount wagered by the respective player exceeds that remaining allocation; and means responsive to the calculations for selectively directing the data representative of some of the calculations to only predetermined ones of the individual stations for display thereat and of others of the calculations to only the central unit for display thereat, whereby information is displayed or not to each of the respective players as an individual or as part of a group to preclude any player from obtaining a competitive advantage over any other player merely by virtue of the displayed information said central unit includes interrupt means for assisting in the selective processing of data from the individual stations on a non-interfering basis by suppressing data entered at any station until data being entered form a time prior thereto from another station is completed, said central unit further includes sequencing means for selectively activating the display at each station to alert the player whose turn it is to make a decision whether to remain in or withdraw from the game, said central unit is responsive to entry to data indicative of withdrawal from the game from all but one of the stations to add the calculation of then-current cumulative amount wagered to the remaining allocation applicable to that one station, said means for selectively directing is responsive to a data entry inquiring as to remaining purchased allocation from any station to direct the calculation thereof made by the central unit applicable to that station for display only at the last-named station, and means operatively associated with said central unit and with each of said stations for selectively designating one of said stations to act as a bank for the purchased allocations.

* * * * *

United States Patent [19]

Koza et al.

[11] **Patent Number:** **4,582,324**

[45] **Date of Patent:** **Apr. 15, 1986**

[54] **ILLUSION OF SKILL GAME MACHINE FOR A GAMING SYSTEM**

[75] Inventors: John R. Koza, Atlanta; Norman T. La Marre, Norcross, both of Ga.; Martin A. Keane, Arlington Heights, Ill.

[73] Assignee: Bally Manufacturing Corporation, Chicago, Ill.

[21] Appl. No.: 567,910

[22] Filed: Jan. 4, 1984

[51] Int. Cl.⁴ A63F 9/22
[52] U.S. Cl. .... 273/138 A; 273/DIG. 28
[58] Field of Search .... 273/1 E, 85 G, 138 A, 273/143 R, 118 A, 121 A

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,735,982 5/1973 Gerfin .... 273/138 A

4,157,829 6/1979 Goldmain et al. .... 273/138 A
4,335,809 6/1982 Wain .... 273/138 A
4,367,876 1/1983 Kotoyori .... 273/121 A
4,494,197 1/1985 Troy et al. 364/412

Primary Examiner—Richard C. Pinkham
Assistant Examiner—Vincent A. Mosconi
Attorney, Agent, or Firm—Jenner & Block

[57] **ABSTRACT**

A video amusement game terminal for a gaming system for playing a game providing the illusion of skill. A game processor provides a video game presentation in response to player control wherein a prize award is disclosed through presentation of achievement by the player of a designated objective. The presentation provides to the player the illusion that the prize award is determined by player skill in achievement of the designated objective.
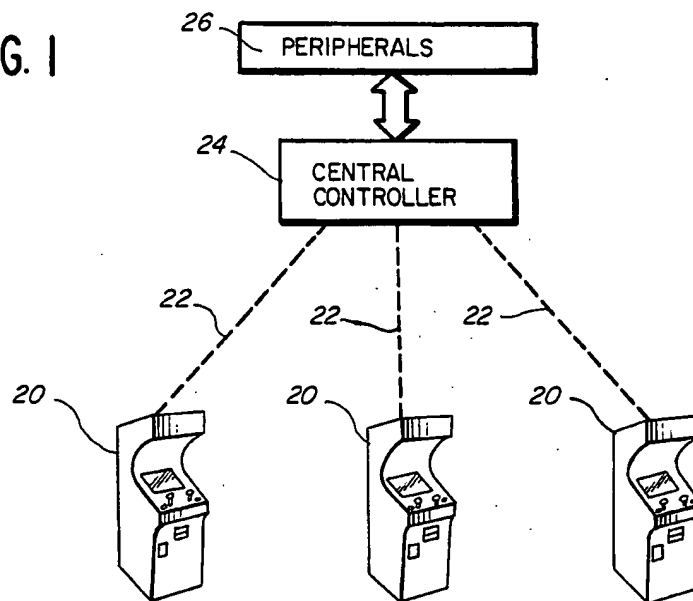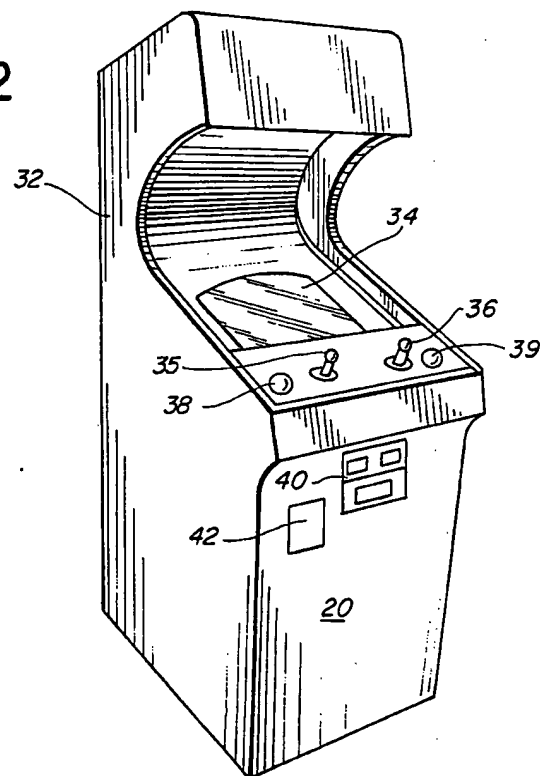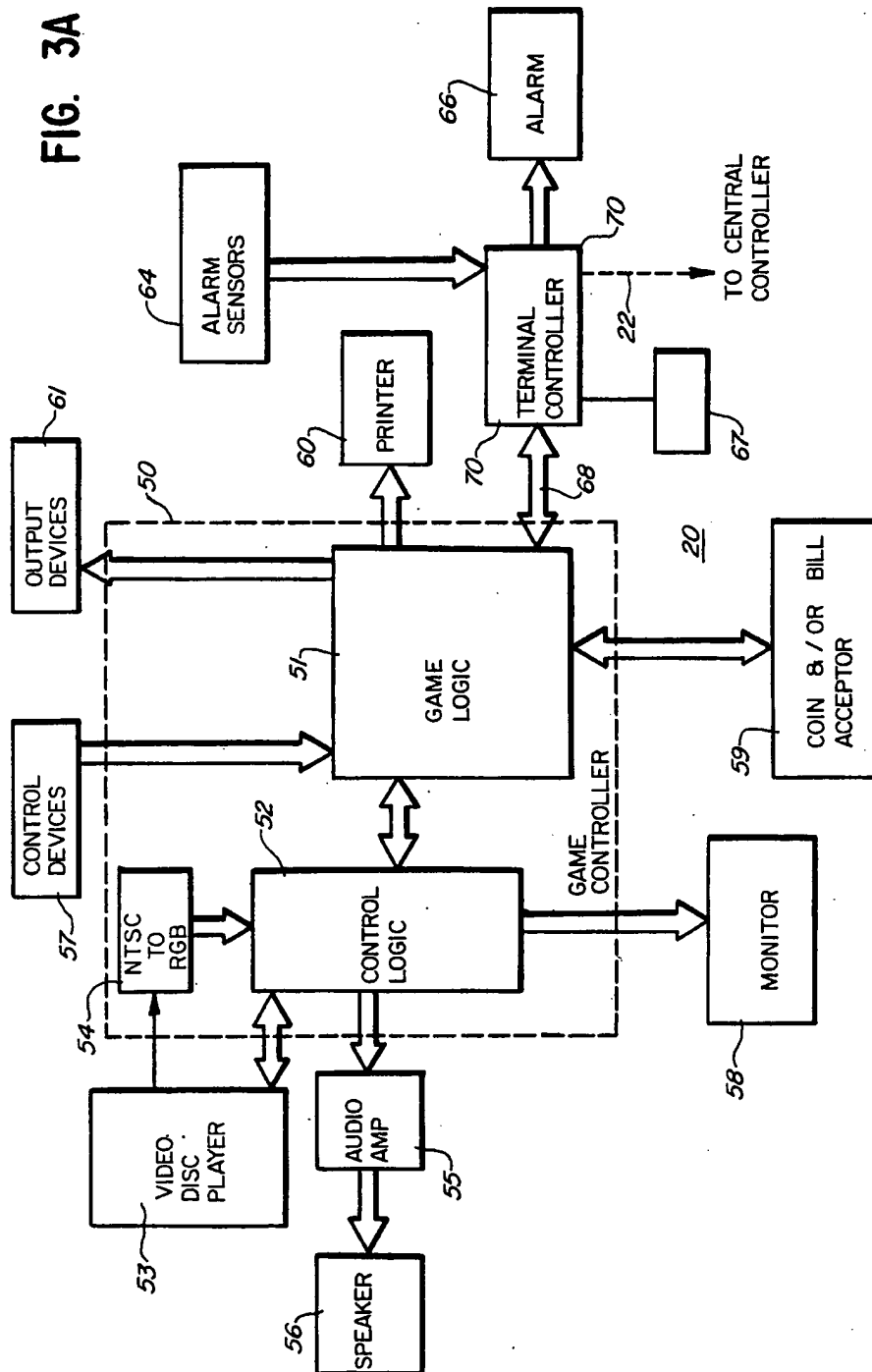
6 Claims, 8 Drawing Figures

FIG. 1

26 — PERIPHERALS

24 — CENTRAL CONTROLLER

22

22

22

20

20

20

FIG. 2

32

34

36

35

39

38

40

42

20

FIG. 3A

FIG. 3B

FIG. 3C

VERT & HORIZ COUNTERS & BACKGROUND CONTROL 110

FORE GROUND GENERATOR 116

VIDEO SELECTOR & DRIVERS 112

VIDEO TO CONTROL LOGIC

117

TO/FROM TERMINAL CONTROLLER

FROM CONTROL INPUTS

TO OUTPUT DEVICES

BACKGROUND MEMORY 108

COLOR RAM 114

CLOCK 118

SIO 120

I/O DRIVER RECEIVER 122

I/O ADDRESS DECODER 124

INPUT BUFFERS 126

OUTPUT DRIVERS 128

92

ADDRESS & DATA BUFFERS 94

DOWNLINE LOAD CONTROL 104

SAFE RAM CONTROL 106

129

51

CPU 100

CTC 96

MEMORY 98

SAFE RAM

BATTERY BACK-UP 102

INTERFACE RAM

ADDRESS DECODER 136

SOUND GENERATORS 138

AUDIO TO CONTROL LOGIC

PROGRAM MEMORY 134

139

CPU 132

130

FIG. 4

FIG. 5

|SYNE|SYNE|MSG NUMBER|MSG LENGTH|MSG TYPE|TEXT|ETX EOB|CRC-16|

|←————————————— MAX 256 BYTES —————————————→|

FIG. 6

1

ILLUSION OF SKILL GAME MACHINE FOR A GAMING SYSTEM

This invention relates generally to electronic gaming systems and more particularly to a video amusement game terminal for playing a game providing the illusion of skill.

Prior art lottery gaming systems have typically involved a drawings or instant "rub-off" games. The current lotteries involving drawing, incorporate computerized systems using electronic terminals operated by licensed retail lottery vendors to dispense printed tickets having the players own selection of lottery numbers printed thereon. These systems are limited in that they require a clerk to operate the lottery terminal. Further, lotteries have been criticized for their appeal to lower income consumers. However, in recent times, there has been explosive growth in the coin-operated video amusement game market which involves an entirely new group of consumers in a market different from any existing lottery. A synthesis of these two fields presents an opportunity to permit consumer operated terminals offering the opportunity to win a prize and to appeal to a new market. However, such a lottery system would preferably utilize a microprocessor based game system which can be customized for different types of games. Since most lottery systems do not allow skill games, a need exists in a video amusement game lottery for a game which provides the illusion of skill in order to attract the largest possible market.

It is accordingly an object of this invention to provide a video amusement gaming machine having video amusement games which provide the illusion of skill.

Briefly, according to one embodiment of the invention, a video amusement game machine for use in a gaming system is provided, having game features controllable by a game player. The machine comprises player control means for providing player input signals for player manipulation of at least a portion of the game features to achieve a designated objective and means for providing an opportunity to win a prize in response to activation by the player and for determining a prize win independent of player input signals. Game processor means provide a presentation of game features according to a predetermined set of game operational conditions, the presentation of game features disclosing a prize award through presentation of achievement of the designated objective and provides to the player the illusion that the prize award is determined by player skill in manipulating game features.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the present invention which are believed to be novel are set forth below with particularity in the appended claims. The invention, together with further objects and advantages thereof, may be understood by reference to the following description and taken in conjunction with the accompanying drawings.

FIG. 1 is a generalized functional block diagram of a specific embodiment of a secure lottery video game with secure remote communications.

FIG. 2 is a perspective view of a specific embodiment of a secure lottery video game remote terminal.

FIG. 3A is a detailed block diagram of a specific embodiment of a secure lottery video game remote terminal.

2

FIG. 3B is a detailed functional block diagram of a specific embodiment of the control logic shown in FIG. 3A.

FIG. 3C is a detailed functional block diagram of a specific embodiment of the game logic shown in FIG. 3A.

FIG. 4 is a detailed block diagram of the specific embodiment of the terminal controller shown in FIG. 3.

FIG. 5 is a diagram of a specific message block format utilized in the illustrated embodiment.

FIG. 6 is a detailed block diagram of the specific embodiment of the central controller shown in FIG. 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 illustrates in functional block diagram form a specific embodiment of a secure video amusement gaming system with remote secure communications. The system is a lottery system which includes a number of player operated remote secure video amusement game lottery terminals 20 (hereinafter referred to as remote terminals). In a preferred embodiment, thousands of such remote terminals 20 may be installed at remote sites such as bars, clubs, retail outlets, etc. It should be noted that this system of terminals may also be used in applications not involving a payed lottery, for example, in retail outlets where play and an opportunity to win a prize is provided to patrons as a promotion (e.g. using a token).

Each remote terminal 20 has the general appearance and functions of a video amusement game including, for example, video display screen, color graphics, sound, laser disc technology, digital video technology, and other video technology. A player activates the remote terminal 20 by placing his money or other form of payment in the terminal and then plays a video amusement game. The player is provided with the amusement and entertainment of a video game while at the same time playing the lottery thus having the opportunity to win a lottery prize. This permits a lottery system with consumer operated terminals and avoids the need for a trained terminal operation clerk. Thus, the remote terminals 20 are not bet collection devices but rather player operated lottery game machines offering the player an opportunity to instantly win while interacting with the game machine. Each remote terminal 20, in the preferred embodiment, permits the player to have a choice of one of a plurality of different video games. The choice of games may include games using only computer generated video, games using only prerecorded video, such as on a video disk, or a game utilizing a combination of both.

Each remote terminal 20 is coupled, as shown, by a communications medium 22 to a central controller 24, which is primarily comprised of a computer. In the preferred embodiment, the communications medium 22 is a telephone link whereby central controller 24 can maintain two-way communications with the remote terminals. Clearly, other communications media will be obvious to those skilled in the art, for example, a two-way dedicated cable, a radio frequency channel, etc. The two-way communications between the remote terminals is conducted using encrypted information so as to prevent unauthorized access to the communicated data.

The central controller 24 maintains supervision over the entire network of remote terminals 20 handling, for example, accounting, validation, security, and seeding

3

of pools, among other tasks. The central controller 24 is coupled, as shown, to a number of peripheral devices 26 such as magnetic disks for storage of data, terminals for operator supervision, and line printers, etc.

Referring now to FIG. 2, a perspective view of a specific embodiment of the remote terminal 20 embodying various aspects of the present invention is shown. Remote terminal 20 is comprised of the cabinet housing 32, having an appearance similar to an arcade video game which contains all necessary electronics. A display screen 34, which may be tilted at an angle as shown, is provided for convenient player viewing. In the illustrated embodiment, display 34 is a raster scan display which permits display of video images, instructions, game rules, odds of winning, and other information. The housing 32 also contains two speakers at the level of the player's ear (not shown) so as to be easily heard and to provide for stereo sound. Player control means are provided as a source of player provided stimuli for transfer to the system electronics. The player control means are mounted below the video display screen 34 and include two joysticks with triggers 35, 36 and two pushbuttons 38, 39, such as are commonly used in the video game art. In the illustrated embodiment, the joysticks 35, 36 provide signals responsive to user movement of the joysticks in a 360° radius about the center pivot point of the joystick to control interaction of the player with the game play and logic. Typically, the joysticks control movement of some "control spot" on the screen such as a dot, cursor, star, arrow, or game character to a desired horizontal and vertical position. The two joysticks allow for dual play while the trigger allows for easy control during fast game play action. The two pushbuttons 38, 39 provide for selection of options such as one or two player game operation and other control functions. Clearly, other player control configurations known in the art may be utilized, such as, touch screens, light pens, mice, audio speech recognition units, keyboards, etc.

Each remote terminal 20 is also provided with a conventional electromechanical or electronic coin mechanism 40 on the front of the housing 32 to accept user coins or tokens to actuate the terminal 20. An optional bill accepter or second coin mechanism may be provided for the convenience of the player and minimize machine down time by providing a second means for payment. Other payment mechanisms may also be used, for example, tokens, debit cards, credit cards, etc. An escrow function is provided such that a player may decide not to play after inserting the money and may then receive cash or a credit receipt in return. Additionally, fewer or greater number of joysticks and pushbuttons, or other player control devices, can be provided according to the requirements of the video games desired.

A printer mechanism is provided at the front of the housing 32 mounted inside the housing. A recessed box 42, mounted behind the front panel of the housing 32 and opening to the front, keeps the printer beyond the player's reach and is used as a receptacle for tickets dispensed from the printer. The printer permits the issuance of lottery tickets as a receipt indicating a lottery win or loss, and permits printing of agent invoices, etc. The terminal housing 32 also includes an access door (not shown) to permit access by authorized personnel to the interior of the housing 32.

The operation of the remote terminal 20 may be understood by reference to FIG. 3A which shows a func-

4

tional block diagram of the remote terminal electronic system. A primary subsystem of the remote terminal 20 is the game controller 50 which includes game logic 51, which performs all functions necessary to control the game according to stored operational conditions. The game controller 50 also includes control logic 52 which controls a video disc player 53 (e.g. an Hitachi 9500SG laser disc player) for playback of high resolution video signals prerecorded on a video disc. Other video recording devices may also be used for storage of prerecorded video signals, for example, digital video systems, videotape, etc. In addition, the control logic 52 couples audio signals to a set of speakers 56 through an amplifier 55, and couples video signals to the monitor 58 under control of the game logic 51. The game logic 51 is coupled, as shown, to a terminal controller 70, and in conjunction with the terminal controller 70, executes all necessary lottery functions. The game controller 50 is further comprised of a NTSC to RGB converter 54 which converts NTSC video signals from the video disc player 53 to RGB signals for application to the video monitor 58.

Player stimulus signals are output from player control devices 57 responsive to player activation of the player control devices, such as the joysticks 35, 36 or pushbuttons 38 or 39 of FIG. 2. The player stimulus signals are coupled to the game logic 51, as shown. In addition, activation control signals are coupled from the coin/bill accepters 59 to the game logic 51 to initiate game play responsive to player insertion of the proper amount. In the illustrated embodiment, a commercially available high speed electro-sensitive rotary (e.g. SCI Systems, Inc., Model 1080-2A) printer 60 is coupled to the game logic 51 to permit printing of lottery tickets under control of the game logic 51. The metallic coated paper used by this type of printer minimizes the risk of alteration and counterfeiting. In the case of high tier winners, the printing of a winning ticket is controlled by a valididation signal from the central controller. A number of output devices 61, such as lights and solenoids, are coupled to the game logic 51, as shown.

The game controller 50 has multiple game selection capability. In the preferred embodiment, one of four games may be selected by the player through the player control devices 57 after activation of the remote terminal by payment of the required fee through the coin/bill acceptors 59. The game controller 50 can operate any of a wide variety of games including real time computer generated video graphics games, games utilizing only prerecorded video signals recorded on a video disc, and games combining real time computer graphics with prerecorded video.

Game programs for such computer graphics games and video disc games are known in the art. The computer control programs involved are stored in program memory and can be downloaded from the central controller 24, through the terminal controller 70 to the game controller 50 to permit changes in the games. There are a wide variety of games which can be utilized.

Computer graphics games which are suitable for the lottery remote terminals include such well-known skill based games as TRON. In a lottery system wherein prize awards are not to be based on skill, these games may be used, for example, by matching a random number to the score generated by the game play. As an alternative, a win or loss determined at the time of activation may be disclosed at some point in the game.

5

Another highly suitable game is a non-skill game such as Lady and the Tiger which is disclosed in detail in a copending application, filed Aug. 11, 1983, by Martin A. Keane, et al., entitled "Video Gaming Machine and Method Based Upon A Dramatic Narration." In this game the player positions a character before a particular door and once he chooses a door, certain events result, such as appearance of a Lady signifying a win, a Tiger signifying a loss, or an open doorway which leads to the next scene. False clues are sometimes used to enhance play characteristics, for example, the Lady's handkerchief appears in front of a door but may or may not indicate the presence of the Lady. In the preferred embodiment, the win or loss is determined when the remote terminal is activated based upon a pool (to be described in detail hereinafter) for that game. .

A novel game approach particularly suitable for the instant lottery system creates the illusion of skill while still determining the outcome at the time the game is started. In a specific embodiment, the player of the game is flying in a spaceship with a cockpit point of view having a selected number of missiles available. The view shows many space objects in the background, occasionally one of the space objects starts becoming larger and moves toward the player becoming a spaceship which is moving rapidly. The player, using the joystick, aims and fires a missile at the spaceship. A "hit" will result in an explosion with a prize value displayed, and shown on a panel on the screen. The process is repeated with a miss merely using up a missile while a hit produces another prize value display. Each missile could follow a curved path which adds realism and the spaceships fly by at varied speed. The player must aim correctly to hit a spaceship, however, as the game proceeds the tolerance for a hit is widened and the explosion of the missile gets stronger to increase the chance of getting a hit. In addition, the number of spaceships increase until it becomes hard to miss. Thus, the level of skill diminishes as the game progresses, as opposed to the conventional practice of increasing difficulty as the game progresses. This gives the player the illusion that a win or loss is dependent upon his skill. In actuality, the game controller completely controls the outcome, such that three identical prize values produce a win of that value. The result, in the preferred embodiment, is determined at the time the game is started based on a pool.

Another type of game that is highly suitable for the instant lottery system is an interactive video disc based game using video image signals and audio signals prerecorded on a video disc. This type of game provides elevated levels of sophistication, quality and realism. Such games, using prerecorded video combined with real time computer graphics, utilizing game control circuitry as shown in FIG. 3A, are known in the art.

A novel game utilizing only prerecorded video signals is particularly suited for the lottery system. In this game, 2 to 5 second sequences (i.e. 60–150 coherent action frames) of "live" game action are filmed and recorded on a video disc. Some computer generated graphics may also be recorded on the disc as well. Appropriate audio sounds are also recorded on the disc in conjunction with the video signals. The prerecorded sequences and audio are manipulated to form game sequences. The computer graphics, either real time or prerecorded, are occasionally inserted between sequences to fill the video disc player search time.

6

In a specific embodiment, the game shows a motorcycle ridden through the the hilly streets of San Francisco, with the player controlling the motorcycle by moving the joystick right or left to turn it. The object of the game is to catch a limousine and obtain a prize value indicator from the passenger. This is done several times, and a win results when three identical prize values have been obtained. False clues are used to enhance play features, for example, the limousine may be shown ahead of the motorcycle turning a certain direction, thus giving the impression that turning the motorcycle in that direction will improve the chances of catching the limousine. In addition, while the game is played, the audio provides the sounds of a motorcycle and distinctive San Francisco street sounds. The player thus can experience some of the sensations of a high speed motorcycle ride through the streets of San Francisco while playing a lottery.

The terminal controller 70, shown in FIG. 3A, is a separate, secure unit within the remote terminal housing 32 which controls all communications in a secure manner to and from the central controller 24 (see FIG. 1), and handles non-secure communications with the game controller 50. The terminal controller 70 also controls all security functions for the remote terminal 20, directs the printing of tickets, and stores terminal play history and financial data for transmission to the central controller 24, and permits downloading of program code and other critical data from the central controller 74.

In addition, the terminal controller 70 makes the determination for the low tier prize awards and controls the awarding of all prizes. The win values are kept in the terminal controller 70 and are sent to the game controller 51 whenever a game starts. If the terminal controller 70 determines a high tier winner is to occur, it initiates a call to the central controller 24 for validation.

Detailed records of play information are kept by the terminal controller 70 as well as records of exact prize payouts and play frequency data on an hourly basis (i.e. hourly meters). This data is battery backed to safely preserve it and is occasionally transmitted to the central controller 24. The terminal controller 70 includes a central processor for control and processing, a modem for telephone line communications, battery back-up and protection circuitry to deter unauthorized access to its stored information. As shown in FIG. 3, alarm sensor signals are coupled to the terminal controller 70 from alarm sensors 64, located in the remote terminal housing and the premises at which the remote terminal is installed to detect attempts at unauthorized access. In addition, alarm signals generated by the terminal controller are coupled to an alarm 66 as well as to the central controller 24 via the communications medium 22. A cash box lock control solenoid 67 is also provided, as shown, to permit control of the cash box. The terminal controller 70 can partially shut down to a power-down, or sleep, condition to save power, and is powered-up either by a call from the central controller 24 or an internal wake-up signal. The communications medium 22, which in the preferred embodiment is a telephone network, links the remote terminal 20 to the central controller 24 to permit, inter alia, a detailed accounting of terminal activity upon request from the central controller 24. In addition, the terminal controller 70 can initiate and transmit exception messages to the central controller 24 for exceptional conditions, such as unauthorized entry, power outage, etc. Communications

7

8

between the terminal controller 70 and the game controller 50 is performed via the cable 68 coupled as shown.

Referring now to FIG. 3B there is shown a detailed block diagram of the control logic 52 of FIG. 3A. The control logic 52 in communication with the game logic 51, controls the video disc player 53, and switches audio output between audio from the video disc player 53 and audio from the game logic 51. The control logic 52 interrogates the video disk player to determine the current frame location and provides instructions to step the player to the next required location. In addition, the control logic 52 switches between the video signals from the game logic 51 and video signals from the video disc player 53 coupling the selected video to the monitor 58.

During play of a game, player control signals (e.g. representative of joystick position) generated by player manipulation of the player control devices 57 are coupled to the programmed game processor 100 of the game logic 51. The game processor 100 generates appropriate graphics, audio and control signals determined by the game operational conditions. These signals are coupled to the control logic 52 where the programmed processor 71 processes the signals and couples them to the monitor 58 and speakers 56 through the video switch 88 and audio switch 90.

For a video disc game the signals from the game logic 51 are primarily control signals which are based upon the game operational conditions. The processor 71, responsive to the current video frame location on the video disc and to the control signals, generates addressing signals to direct the disk player 53 to the next required video frame location for playback of the appropriate video and/or audio signals. The addressing signals are determined using a table or algorithm which relates the player control inputs to the vidoe image addresses consistent with the game operational conditions. The video and/or audio signals from the video disc player 53 are coupled, under control of the processor 81, to the monitor 58 and the speakers 56 through the video switch 88 and audio switch 90.

The control logic 52 comprises a central processor 71 (e.g. a Zilog Z-80) which is coupled to other circuits, as shown, via a universal bus 72. A memory 74 is provided for program and data storage, comprising read only memory (e.g. Intel 2764) and random access memory (e.g. Hitachi HM6264P). A conventional clock generator 76 is provided to generate required system clock signals, and a counter-timer chip 78 (e.g. a Zilog 8430) provides timing, interrupt, and control signals, with each coupled, as shown, to the bus 72. A serial input-output (SIO) circuit 80 (e.g. Zilog 8440) is coupled, as shown, to the bus 72 to provide serial to parallel, and parallel to serial conversion to interface with the I/O circuits 82, 84. The I/O circuit 82 is an RS232 Driver and Receiver (e.g. Motorola 1488, 1489) which couples output signals to the game logic 51. The I/O circuit 84 is also on RS232 Driver and Receiver for coupling control signals and address signals to the standard control input of the video disc player 53. An Input/Output Decoder circuit 86, comprising a conventional decoder (e.g. 74LS138) and a latch (e.g. 74LS273), couples to the bus 72, as shown, primarily to provide decoded control signals from the CPU 71 to a video switch 88 and an audio switch 90.

The video switch 88 (e.g. Motorola 4066's), based on control signals generated by the CPU 71, switches

video signals from either the game logic 51, or the video disc player 53 to the monitor 58. Thus, when computer generated graphics are to be displayed, the switch 88 couples the signals from the game logic 51 to the monitor 58, and when prerecorded video is to be displayed, signals from the RGB converter 54 are coupled to the monitor 58. Similarly, the audio switch 90 (e.g. Motorola 4016) switches audio signals from either the game logic 51, or the video disc player 53 to the audio amplifiers 55 and thus to the speakers 56, under control of the CPU 71. The audio switch 90 is capable of controlling each channel of stereo sound separately, thus, one channnel from each source may be simultaneously coupled to the audio amplifiers 55.

In FIG. 3C, there is illustrated a detailed block diagram of the game logic 51. A central processing unit 100 (e.g. Zilog Z-80) is coupled, as shown, to a bus 92 through a conventional address and data buffer 94 (e.g. Motorola 74LS245, 74LS244) The CPU 100 is also coupled to a counter-timer circuit 96 (e.g. Zilog 8430) which generates interrupt signals and to a memory circuit 98, as shown. The memory circuit 98 provides storage for program code and data, and is comprised of read only memory (e.g. Intel 2764, 27132) and random access memory (e.g. Hitachi HM6264) which is protected by a battery backup circuit 102. Part of the random access memory is safe RAM, which is protected by a security program, for safe storage of important machine meter data. In addition, part of the RAM is accessable, as shown, to downline load control circuitry 104 to permit downline loading of new program code and data. The safe RAM is controlled by the CPU from the bus 92 through a safe RAM control circuit 106. The bus 92 couples, as shown, to video generation circuitry composed of a background memory 108, vertical and horizontal counters 110, a color RAM 114, a video selector and driver circuit 112 and a foreground generator 116. The video generation circuitry creates video signals based upon control signals from the CPU 100 which specifies the images to be displayed and their locations on the screen. The monitor 58 ultimately converts the signals to visible game and diagnostic images. A detailed description of the video generation circuitry is disclosed in a co-pending application filed Aug. 8, 1983 by John J. Pasierb, et al., Ser. No. 520,762, and assigned to the instant assignee. A video output 117 from the video driver circuits 112, couples video signals to the video switch 88 of the control logic 52 (see FIG. 3A).

A conventional Serial Input-Output (SIO) 120 circuit is coupled, as shown, to the bus 92 to provide serial to parallel and parallel to serial conversion for interfacing to the I/O circuits 122. The I/O circuits 122 comprise conventional RS232 Driver and Receiver circuits for communications with the terminal controller 70 on cable 68. A baud rate clock 118 provides clock signals to the SIO circuit 120. A conventional Input/Output address decoder circuit 124 (utilizing, e.g. 74LS138, 74LS139) is also coupled to the bus 92 to provide address decoding for the SIO 120, and for input buffers 126 and output drivers 128. The input buffers 126 couple signals from the player control devices 57 through the bus 92 to the CPU 100, and signals from the printer 60 and the coin/bill acceptor 59. The output drivers couple signals from the CPU 100 through the bus 92 to control the output devices 61 and other outputs, such as printer 60, lights, solenoids, etc.

In addition, a sound generator system 130 is coupled to the bus 92 through an interface RAM 129, as shown. The sound generator system 130 primarily comprises a central processing unit 132 (e.g. Zilog Z-80), coupled to a bus 139, with conventional program memory 134 (containing ROM and RAM) and address decoding 136 also provided, as shown. The CPU 132 controls two programmable sound generators 138 (e.g. General Instruments AY-3-8912) to produce highly complex sounds upon requests from the CPU 100, which are coupled to the audio switch 90 of the control logic 52 (see FIG. 3A).

Referring now to FIG. 4, there is illustrated a detailed block diagram of the terminal controller 70 shown in FIG. 3. A two-way communications line 22, is coupled from the central controller 24 through a conventional input protection circuit 140 to a modem 142 (in the preferred embodiment a 300 baud, auto-answer, auto-dial modem chosen for cost effectiveness), as shown, to permit reception or transmission of encrypted data. A ring detection circuit 141 detects an incoming call and couples a detection signal to a wake-up circuit 178, as shown, thereby activating the wake-up circuitry in the event that the terminal controller 70 is in a power-down state. The wake-up circuitry generates an actuation signal which is coupled, as shown, to a battery backed sleeper power supply 170 providing power to a terminal controller processor 150.

The modem 142 is coupled through a conventional serial communicator 144 to the terminal controller processor 150, as shown. In addition, the cable 68 is coupled from the game controller 50 (see FIG. 3) through conventional input protection circuitry 146 to a standard RS232 interface 148 to permit communications between the terminal controller 70 and the game controller 50. The interface 148 couples signals through the serial communicator 144 (in the preferred embodiment, comprising Zilog 8440 SIO's) to the terminal controller processor 150, as shown. The processor 150 is preferably a low power processor, such as a Zilog Z-80, and is coupled directly to a program memory 152, as shown. The memory 152 includes ROM and battery backed up, down-loadable RAM which permits alteration of program code from the central controller. In addition, a battery powered RAM 154 is provided to permit safe data storage. A battery powered encryption and decryption key RAM 156 permits storage of an encryption or decryption key used for encrypting or decrypting data for secure communications between the terminal controller 70 and the central controller 24. The encryption key stored in encryption key RAM 156 is utilized by the processor 150 to encrypt any information which is to be transferred from the terminal controller 70 to the central controller 24 and the decryption key is used for decrypting received data from the central controller 24. The encryption and decryption key RAM 156 and RAM 154 is also coupled to a memory battery 162 through a "key killer" circuit 158 which cuts off power to the RAM 156 and 154 in response to an attempt at unauthorized access to the terminal controller 70, thereby erasing the encryption and decryption key and all other RAM data. The unauthorized access is detected by sensors 163 coupled to a terminal controller internal alarm 164 which generates an alarm signal which is coupled as shown to the key killer 158.

The power line, constituting the primary power source, is coupled as shown to a battery charger 166. The battery charger 166, is coupled to a battery 168,

and thus continuously charges this battery to provide for battery back-up during power line failures. The sleeper power supply 170 is coupled to the battery 168 to provide short-term back-up power for the terminal controller processor 150. The memory battery 162 is coupled, as shown, to the battery 168, and to the sleeper power supply 170, both of which charge the memory battery 162. The battery 168 also provides power to the alarm circuit 172, which generates an alarm signal in response to the detection of unauthorized access to the remote terminal housing 32 by any of a number of sensors 64. The alarm signal is coupled to the terminal controller internal alarm circuit 164, the terminal controller processor 150, and to a protection circuit 174 which drives an alarm transducer 66 (e.g. bell, light, etc.). The alarm sensor signals are coupled through a conventional protection circuit 176 and through a sleeper circuit 178 to the alarm circuit 172. An alarm sensor signal causes the sleeper wake-up circuit 178 to activate the sleeper power supply 170, if the system is in the power-down mode. Normally, the processor 150 determines if the alarm 66 should be activated and inhibits the alarm 66, if it should not be activated. If the processor 150 fails to activate, the alarm 66 will not be inhibited. The processor 150 also determines if a call to the central controller is required and initiates the call when appropriate.

The terminal controller circuitry described hereinabove, incorporates numerous novel security features to provide for secure functioning of the remote terminal. In addition to battery back-up and protection circuitry to permit erasing the encryption keys, the terminal controller 70 is enclosed in its own tamper resistant enclosure with sensors 163 attached to detect tampering. Other sensors 64 are positioned within the remote terminal housing 32 and at all doors to detect penetration of the housing 32. The terminal controller 70 includes door sensors, AC power sensors, and phone line connection sensors. In addition, the terminal controller 70 allows for connection to the burglar alarm of the premises. The terminal controller 70 can also relay a message indicating detection of tampering or power failure to the central controller as well as to react locally with alarm 66, shut-down of the remote terminal 20, or erasure of secure information. All communication between the terminal controller 70 and the central controller 24 are sent in encrypted form.

A public key encryption method known as RSA public key encryption can be utilized as a high security method of encryption. This scheme involves the use of a secret key by each controller for decryption. The encryption key is made common knowledge but the decryption key is not available and is kept secure in the decryption key memory 116 of the terminal controller 70. This RSA method is known to be computationally intractable. This method may be used only for highly sensitive data such as transmission of medium security encryption keys or high tier winners because it is computationally complex and time consuming.

A second encryption method which may be used is the widely known DES (Date Encryption Standard) method developed by the National Bureau of Standards. This method is used for higher speed encryption of medium security data, such as low security encryption or decryption keys or seed data. To improve security, multiple levels of DES encryption may be used. Simpler, but less secure encryption methods which are known in the art are options for low security, high

11

speed data transmissions, such as meter data, or program code.

A standard message format is utilized for transmissions between the terminal controller 70 and the central controller 24, as illustrated in FIG. 5. A block with a maximum message length of 256 bytes of eight bits each is used. As shown, the first two bytes are synchronization codes, followed by a message number, then the message length. The message number is an eight bit number which is incremented after each message is sent, and is used for determining error recovery on missing blocks as well as for security to detect unauthorized communications. The message length is the number of bytes used for the text segment of the block with a maximum of 247 bytes per block. This text segment is encrypted for all transmissions. Following the message length is a message type code, with four types used in the illustrated embodiment. They are (1) STX-data to follow, (2) ACK-previous message correctly received, (3) NAK-previous message not correctly received, and (4) XOF-retransmit previous message after ten seconds. After the text bytes, an end of text or end of block code is inserted followed by two bytes of cyclic redundancy code and a guard end byte. If a message is more than 247 bytes, multiple blocks are used for the message.

Among the secured functions performed by the terminal controller processor 150 is the determination of prize pools to insure accurate prize payouts guaranteeing a predetermined total prize value within a preselected group of plays, referred to as a "pool." In a preferred embodiment, each pool contains a million plays and each pool is further divided into a thousand mini-pools of a thousand plays each. This multi-level pool system may be expanded to three or more levels as needed. For example, a major pool for high tier winners, secondary pool for intermediate winners and a mini-pool for low end winners.

The mini-pool is used to implement a fixed low-end prize structure such that within a small number of plays (i.e. 1000) there is a preselected amount of low-end prize value (for example, low-end winners could include all prizes of $25 or less). Thus, there is a known amount of low-end prize value in each mini-pool. This ensures that each terminal will have its share of winning value and may be accomplished by using a predetermined number of each win value or by a distribution of different wins having equal total prize values. The central controller 24 assigns a mini-pool for each particular game to each remote terminal 20 from its larger supply of 1000 mini-pools. Since each remote terminal 20 has a number of different games, each terminal controller maintains an equal number of separate mini-pools. Each time a mini-pool is completely used, the terminal controller 70 initiates a telephone call to the central controller 24, reports on the completion of the mini-pool, and a new mini-pool is assigned by the central controller 24. When all 1000 mini-pools of a given pool have been assigned, the central controller 24 then creates a new pool.

Intermediate and large or grand prize winners are generally not sufficient in number to permit even one of them in each mini-pool. Thus, there is a selected set of these high tier winners in each pool, and some mini-pools will contain such a large prize while some may contain none.

The pools and mini-pools are determined based on random seed numbers. In the preferred embodiment, pool seeds, and minipool seeds are used which determine the outcome of every play. The digits for the pool

12

seeds are created at the instant when the central controller needs to start a new pool. These digits are generated using a pseudo-random number generator, such as are known in the art, using an input from the computer's clock. As a result, there is no way to predict in advance the randomization of a given pool.

When a mini-pool seed is requested by the terminal controller 70, a value from the terminal controller clock is transmitted with the request. The digits of each mini-pool seed are generated at the central controller 24 using the pool seed and the terminal controller clock value to feed a random number generator. The resulting mini-pool seed is then transmitted to the terminal controller 70. In addition, a high tier win is reserved to be included in the mini-pool. This seed is then used to randomize the mini-pool thus determining which players win both high tier prizes and low-end prizes. This is done by using the random seed to shuffle the wins into a random order based on a randomizing algorithm, such as are known in the art, so that the distribution of wins within each pool will not be predictable. A suitable randomizing algorithm is the linear congruential method or a multiple pass variation of this method. The random number seeds and the randomizing algorithm are also stored by the central controller 24 to permit the central controller 24 to recreate the distribution of wins for use in validation. It should be noted that the winners in the games can also be generated using other methods, for example, where each individual play on the remote terminal is an independent uncontrolled random event.

The terminal controller 70 is also involved in the validation process of high tier wins. When the terminal controller 70 determines that a high tier win will occur for the play of the game currently started, the terminal controller calls the central controller 24 to request validation. The central controller 24 checks to verify that the high tier win is correct and that the allotment of high tier winnings for that terminal has not been used up. If the win is valid, the central controller 24 will transmit a win amount and validation code to this terminal controller 70. The terminal controller 70 sends this information to the game controller 50 for printing on a lottery ticket. The validation code on the ticket can be used to verify authenticity prior to payment. If the high tier prize was not correct the central controller 24 can send a shut-down message to the terminal controller 70, which then shuts down the remote game terminal. If communication between the terminal controller 70 and the central controller 24 cannot be made before the end of a game, the terminal will not print a winning lottery ticket but instead the remote terminal notifies the player that he has won a prize and prints a special ticket that he must take to a validation center to determine the value of the win. If the printer is disabled, the terminal will void the play and the player will obtain a refund. The terminal controller 70 can then send a message to the game controller 50 disabling the game until communication can be established.

In an alternative mode, a high tier winner may enter his name and address, or other information into the remote terminal 20 using the player control devices 57. The terminal controller 70 then transmits this information to the central controller 24 thus electronically registering the winner. The game controller then prints a receipt but the winner need only go to a claim center to obtain payment. This electronic claim process improves the security of the claims process.

13

Terminal controller 70 serves also as a storage location for collection of various remote game terminal meter data, such as, the number of games played, money in the terminal, the amount of winnings paid out, and other data. The terminal controller 70 periodically polls the game controller 50 to make sure it has up-to-date meter data. The hourly meter data is kept by the terminal controller 70 in battery backed RAM 154 until it is requested by the central controller and is then transmitted. Once the central controller 24 has received the data it sends a message back to the terminal controller 70 that tells it that the meter data is safely stored in its memory, and that the terminal controller storage is freed up for reuse. The integrity of the meter data is also maintained across the telephone transmission lines through the use of a cyclic redundancy code (e.g. CRC-16 method as used in the IBM binary synchronous communications protocol). The addition of the cyclic redundancy code to the data is a commonly used method for verifying the correctness of transmitted data. If an error is detected at the receiving end, the central controller 24 sends back an error message to the terminal controller which then retransmits the data until it gets through correctly.

Additional security features of the terminal controller 70 involve detection and transmission of exceptions such as alarms, printer empty, malfunctions, power failure, and others. When an exception is detected, the terminal controller 70 determines the severity of the exception and sends a message to the central controller for the more serious exceptions while logging the less severe exceptions for later transmissions.

The program set for the terminal controller uses a commercially available operating system for linkage of the various tasks operating within the terminal controller. This operating system is called AMX, for "A Multitasking Executive" (marketed by KADAK Products, Limited). It allows for up to 63 concurrently running tasks of various priorities to send messages and schedule other task on a real time interrupt driven basis. It includes user tasks, such as real time clock/calendar, task scheduling on a time of day or a periodic basis, buffer management, intertask communication with buffering, and many other features. Under the operating system there are three main types of programs: restart procedures, interrupt service procedures, and user tasks. The terminal controller also has a resident library of commonly used re-entrant subroutines.

The scheduling algorithm of the operating system for task execution is absolute scheduling, so that the highest priority task is given all the time that is needed and only when it gives up the processor is the next highest priority task allowed to run. This continues down to the lowest priority task. There are no two tasks with equal priority. This means that a task of high priority will inhibit the operation of all tasks of lower priority.

Within the terminal controller 70, in the preferred embodiment, there are three types of true interrupts and one type of "polled" interrupt. The three true interrupts are the clock interrupt, the I/O character interrupt, and the non-maskable interrupt. The one polled interrupt is the alarm interrupt. The terminal controller 70 in the form of the preferred embodiment, uses the Z-80 mode two interrupts, or vectored interrupts, which allows the interrupt routines to reside anywhere in memory.

The terminal controller 70 generates regular clock interrupts of a programmed frequency, which in a preferred embodiment is 100 hertz. The operating system is

14

set up to count down any integer number of clock interrupts into one system tick time (the basic system time of which all software timers are multiples). For the preferred embodiment of the terminal controller, the 10 milliseconds (100 hertz) clock interrupt is used as the system tick time. At tick time, tasks are reevaluated for priority of execution, all counters are decremented, and the time of day is incremented.

A vectored interrupt mode is used for all external communications. Thus, the terminal controller 70 can automatically jump to various routines. In general, the character input routines will read a character from a port and write it through a circular list, and the output routines will read a character from a circular list, and write it out to a port.

A non-maskable interrupt will occur only when the power has been removed from the terminal controller 70 for long enough for the battery to be low. At that time the terminal controller will initiate a phone call to the central controller advising it of the situation, and will then go inactive.

The alarm interrupt is actually a polled interrupt as it doesn't actually interrupt the processor, but instead is polled during clock interrupts. An I/O port contains various bits indicating various machine statuses such as door open, or power outage. The I/O port also has one bit which is latched whenever the status of that port changes, so a read of that bit during clock interrupt will quickly determine if any actions are required. If no change has occurred in the latched bit, the terminal controller 70 returns to execute the highest priority current task. If the bit indicates that a change has occurred, the interrupt routine stores the alarm condition and schedules an alarm task to be run.

The terminal controller 70 can modify its own programs while it is operating based upon data received from the communication line 22. This is typically used to correct small bugs which are discovered after the machine is in the field. It can also be used to generally enhance machine programs from the central controller 24 or write new game programs from the central controller 24 without requiring direct reprogramming.

To allow for this program downline loading, the terminal controller 70 tasks are defined with several non-existent tasks in the operating system task table. The non-existent tasks are simply defined to the operating system so that their entry points are at some predefined address that initially contains a return instruction. Then tasks are set up so that the non-existent tasks are never scheduled to run. These holes in the operating system task table are of various levels of priority to allow for various priority options to be added.

When a task is to be added, an appropriate section of unused memory is first filled with the new task received from the communications line 22. Next, the operating system task table is modified to include the new task. The new task is scheduled to run by either directly modifying the operating system task table or by setting up conditions that cause the existing task to schedule the new task to run.

Various routines make up the terminal controller software including, the restart routines, the interrupt service procedures, the high priority tasks, and the low priority tasks. The high priority tasks include those that recognize communication, sensor, or clock oriented events, and then pass control to a central message switcher task. The message switcher task then calls the

routine that can process the message involved. This allows for modular software.

Sensor events (e.g. door open, character arrival, etc.), communications events (e.g. messages that arrive on either the communication line 68 to the game controller or the communication line 22 to the central controller), and clock events (e.g. the hourly meter store, the daily meter store, etc.) send messages to the message switcher task which calls various event handler tasks which actually process the events.

Most of the work is done by the event handler tasks, which include routines to determine win values, and to create messages for sending to the game controller 50, or central controller 70. All these routines use a common library of reentrant subroutines. The various tasks are arranged in order of priority with the communication sensor and clock events being either interrupt driven or of extremely high priority, running on a polled basis from the clock interrupt routine. Next in order of priority of tasks is the message switcher, which accepts the request for processing and prioritizing the events, and then calls the appropriate routine to handle each event. The subroutines of the message switcher may determine that other processing must be done to complete the processing of the events, and may reschedule the message switcher repeatedly. The lowest priority tasks are routines which are completely non-critical with respect to time, such as the terminal controller self checking routines which verify the proper operation of the terminal controller 70 and which run when the terminal controller has nothing else running.

A number of restart routines are used that are executed only at system start-up which perform such tasks as device initialization, setting time of day, setting counter timers, and so forth.

When the terminal controller 70 first begins operation at power-up or restart, there is a jump instruction to jump to the AMX operating system entry point. The operating system does initialization and then calls the restart routine. The restart routine sets the processor interrupt mode, programs the input ports and programs the clock interrupts. The restart routine then determines the cause of the restart and calls the appropriate subroutine for that type of restart.

Under the AMX operating system, each interrupt must have its own interrupt routine. The interrupt routines include the following routines.

A clock interrupt service procedure routine first stores register values and then checks the alarm I/O port to determine if an alarm has occurred, and then reads and sets the alarm ports if an alarm has occurred. The program then schedules the alarm handler task routine if an alarm has occurred and then jumps to the standard operating system clock handle routine.

A serial input service routine is used to read in characters in response to a character interrupt and then writes the character to a circular list associated with the input line. This provides buffered input of up to 255 characters on all incoming serial communications lines. After the operation is completed, control returns to the task that was interrupted.

A serial output interrupt service routine is used to check the output circular buffer for a character for each line, and then write the character to the output port associated with the output line if there is character, and returns from the interrupt immediately if there is not. This provides buffering of up to 255 characters on all outgoing serial commuications lines. After the opera-

tion is completed, program control quickly returns to the task that was interrupted. The routine is initiated when a transmit buffer empty interrupt occurs causing the terminal controllers vectored interrupt mechanism to vector to the serial output program. There is one entry point for each serial input line and each entry point sets up registers for line dependencies and jumps to the standard character output handler.

There are a number of functions where a table system is utilized and where a constant data file area is needed (e.g. the central controller's phone number, the encryption keys). Thus, certain data areas are linked into memory to provide the necessary data fields. The data areas include the following:

A custom data area is used to store data that makes one terminal controller different from another, such as the central controller phone number, encryption keys, decryption keys, etc. Another data area is used to store variables that are used by various routines but which are not defined in any other manner. Another data area is used to hold finite state tables that are used to define the operation of the message switcher routine. They are separated from the code to allow for quick changes to be made for the operation of the protocol from the central controller without major changes to the code.

The task routines are stored in order of the priority of execution so that higher priority tasks are always executed before lower priority tasks. The task routines can be described in general as follows:

A clock handler routine is the highest priority task in the system. This task updates the time of day and calls a subroutine called CLKSHED, which checks to see if periodic tasks are to be scheduled. It is the highest priority task because the clock would be corrupted if it did not run each and every clock interrupt.

Another routine T10TSK is used to test each serial line status every 10 system ticks and if it is not transmitting, the routine reads a circular list for a character. If there is a character, the character is written to the output port. If there's not a character nothing is done. The output serial lines are interrupt driven and therefore do not require attention while they are running. However, some method is needed to start the line each time another message is to be sent, thus the need for this routine.

An alarm condition task routine ALMTSK is a routine which determines the type of alarm which has occurred, and then sends a message to the message switcher informing it which alarm has occurred. The task is scheduled by the clock interrupt service routine whenever an alarm condition is detected.

A central controller communications task CCCIRCUIT is used to establish communications with the central controller and controls such things as dialing the phone, exchanging passwords, and terminating the communication in an orderly fashion. A ring detection signal or a flag showing that a call is to be initiated causes entry to this task routine.

A data link layer task routine CCDATALINK, scheduled and terminated by the CCCIRCUIT routine, handles the acknowledge and no acknowledge of messages, the CRC calculation and the block formatting of messages for communications with the central controller.

An incoming message task routine CCINLINK is used for handling the incoming communications from the central controller. This routine waits for detection of two sync bytes then inputs one communication block

and passes its address to the communications data link routine. This routine is scheduled and terminated by CCCIRCUIT and communicates with the CCDATA-LINK routine.

A communications outgoing task routine CCOUT-LINK handles outgoing messages and transmits the two synchronization bytes, message number, message length, message type, the text, the ETX or EOB prefixes, and the CRC code. This routine is scheduled and terminated by the CCCIRCUIT routine.

A game communications task routine COMGAME, scheduled by the message switcher routine, handles all communications bidirectionally with the game controller. Its operation consists of getting the message on the communications line and sending the message to the message switcher routine, or fetching a message from another task and sending the message to the game controller.

A meter task routine MTRSK reads the meter values and stores them for transmission to the central controller the next time the communications with the central controller is established. This task is automatically scheduled every hour by the operating system clock handler.

A message switcher task MSGTSK is a major task within the terminal controller software. Various high priority tasks sense events requiring some response from the terminal controller and send the message switcher prioritized messages that the message switcher must handle. The message switcher looks at the message number of the incoming message, looks up the number in a table of subroutines, and calls the subroutine that handles the message. The subroutine processes the message, sets the various appropriate flags, and occasionally reschedules the message switcher to run with the new message that the subroutine has created. This permits quick modification from the central controller of the tables to alter the operation of the terminal controller with minor or no impact on other routines.

A memory integrity task MEMTSK is the lowest priority task in the system. This task verifies the integrity of the memory and informs the alarm handler if an error is detected.

Also stored within the memory of the terminal controller is a resident library of commonly used routines which are reentrant so that they may be used by more than one task. The resident subroutine library includes the following routines.

A routine called INITLOAD is used to get characters from the game controller interface and load them into RAM. No security is used for this transfer and this routine is used only at system start-up to load the rest of the non-ROM based routines into memory.

A routine called INSTALL is used by the central controller for installation of the terminal controller at the agent's sites. The central controller enters values such as the phone number of the central controller into the terminal controller's memory. Once the terminal controller 70 has successfully finished communicating with the central controller 24, a fuse is blown, completing the installation process and the terminal controller 70 becomes fully operational.

Another routine referred to as the Buffer Manager routine controls access to groups of temporary buffers that may be needed by various tasks from time to time. This routine is part of the AMX operating system.

Another routine called CLKSHED is called once every second by the time/date routine within the AMX operating system. This routine checks the current time to see if any of the tasks that are to run at certain times are to be run at that time and if so schedules them to run.

A routine GETTIME takes the clock/counter values from the clock/calendar circuitry and sets the values in a buffer.

A SETTIME subroutine sets the clock/calendar circuit with the time and date from a time/date buffer.

A GETCC subroutine fetches a character from the circular list, or waits for the next character if there isn't an existing character in the list.

A PUTCC subroutine writes a character to the circular list or waits until there is a location available and then writes the character.

A CRCGEN subroutine is used to generate CRC code characters.

A SAVREG routine is a general purpose register saving routine which saves all registers and flags on the stack and then returns to the position at which it was called.

A RSTREG subroutine is a routine which takes the registers and flags that were put into the stack by the SAVREG subroutine and restores them to the registers.

A POOLGET subroutine takes a pool seed and iteration number as an input and returns a win value after being called.

A DECRYPT subroutine is a general purpose decryption routine that is called whenever decryption is required. It takes an encrypted buffer and decrypts it with the currently active decryption key, utilizing the specified encryption level. The result is placed into a buffer which is then returned to the calling program.

A ENCRYPT subroutine is a general purpose encryption routine. Whenever a buffer needs encrypting, this routine is called with the address of the buffer and the level of encryption to be used. This routine then encrypts the buffer and returns the address of the encrypted buffer to the calling program.

A DIAL subroutine dials the phone to call the central controller.

A ANSWER subroutine answers the phone when the central controller calls.

A library of subroutines for use by the standard compiler is also part of the run time library.

The central message switcher task calls the event handler subroutines on a one at a time basis to process each event as it occurs. There is one event handler routine for each condition to be handled. The event handler subroutines that are used include the following routines.

A MSGPOL routine is called by CCCIRCUIT whenever a poll message is received from the central controller. It interprets the message and creates an appropriate message for transmission back to the central controller. It then schedules the message switcher for the transmission of the message to the central controller.

A MSGMDI event handler subroutine is called whenever a MDI (i.e. marketing and accounting data) message is to be transmitted to the central controller and sends the message to the CCCIRCUIT routine for transmission.

A MSGAOU event handler routine is called whenever an AOU (Acknowledge OUtgoing message) message is received from the central controller. This means that the previous message sent by the terminal controller has been fully processed and the buffer space that is used may now be reused by the terminal controller.

19

A MSGHDR event handler routine is called by the message switcher whenever an hourly meter data request is received from the central controller. It interprets the request and creates an hourly meter data message for transmission back to the central controller, and then schedules the message switcher for the transmission of the MDI message to the central controller.

A MSGHTD subroutine sends the hourly data transmission message to the CCCIRCUIT routine for transmission to the central controller and is scheduled by the message switcher routine.

A MSGDOR routine is called whenever the condition of one of the game doors is changed and sends a door open message to the central controller. This door change condition is detected by the alarm task ALMTSK which schedules the message switcher to run the MSGDOR routine. The MSGDOR routine formats the message and sends it to the CCCIRCUIT routine for transmission to the central controller.

The MSGDRC routine is called by the message switcher in response to a message from the central controller to control the condition of the cash box door solenoid. It can lock or unlock the cash box, depending upon the content of the message.

A MSGINV event handler subroutine is called by the CCCIRCUIT routine whenever an invoice message is received from the central controller. The invoice message consists of the information necessary for the printing of the agent's invoice. The MSGINV routine formats the agent's invoice and prints it. A AIN (Acknowledge INcoming message) message is then scheduled to be sent to the central controller.

A MSGAIN routine creates an AIN message which is sent to the CCCIRCUIT routine for transmission of the AIN message to the central controller. This message informs the central controller that various messages have been correctly received and processed.

A MSGMPR routine is called by the Pool Manager routine whenever one of the minipools is low on win values. A phone call to the central site is then initiated and an MPR message is transmitted.

A MSGMPS event handler routine is called by the CCCIRCUIT routine whenever an MPS message is received from the central controller. This routine updates the minipool values for use in determining future win values.

A MDGBWD event handler routine is scheduled by the message switcher whenever a big winner has occurred and results in a sending of a message through the CCCIRCUIT routine to the central controller.

A MSGBWV subroutine is called by CCCIRCUIT routine whenever a big winner validation message is received from the central controller. This message is then processed and sent to the big winner printer for printing of a big winner ticket.

A MSGECR subroutine is called by the message handler and generates the electronic claim receipt reject message sent by the CCCIRCUIT routine from the terminal controller to the central controller. If the printer fails or communications with the printer is lost, the winner will receive a ticket which must be validated at a later date.

A MSGECD routine is scheduled by the message switcher to send an electronic claim data message through the CCCIRCUIT routine from the terminal controller to the central controller.

A MSGTPF event handler subroutine which is scheduled by the alarm task ALMTSK is a routine

20

which formats a ticket printing failure message which is then handled by the CCCIRCUIT routine to send a message from the terminal controller to the central controller. In case of the printer failure together with communication loss with the central controller, the player will receive his money back.

A MSGCPO event handler subroutine is scheduled by the message switcher in the event of a ticket printer failure and generates a claim printer order message. The message is sent by the CCCIRCUIT routine to the central controller.

A MSGTPA event handler subroutine is scheduled by the message switcher when a message is received to handle ticket printer acknowledge messages.

A MSGEXD event handler subroutine is scheduled by the message switcher when an exception has occurred and sends a message which is an exception data message to the CCCIRCUIT for transmission from the terminal controller to the central controller.

A MSDMSD event handler subroutine is scheduled by the message handler and sends market survey data from the terminal controller to the central controller by formatting a message and handing it to the CCCIRCUIT routine for transmission to the central site.

A MSGCMD event handler subroutine is scheduled by the message switcher in response to reception of a command message from the central controller and passes the appropriate message to the required routine.

A MSGCMR event handler subroutine is scheduled by the message switcher and formats the command response to the central controller in response to command data received from the central controller. The CCCIRCUIT routine then handles transmission of the response.

A MSGPLH event handler subroutine is scheduled by the message switcher in response to a message received from a central site. The routine solicits marketing, accounting (same as MDI) and hourly data from the terminal controller and constitutes the first daily poll. The MSGMD2 event subroutine is then scheduled to run.

The MSGMD2 event handler subroutine generates a message for transmitting the marketing, accounting and the hourly data from the terminal controller to the central controller, then hands the message to the CCCIRCUIT routine for transmission.

A MSGPWD event handler subroutine is scheduled by the message switcher in the event of a printer failure. If the printer fails or power goes down, the remote terminal terminates play before informing the player of a win and returns money to the player.

FIG. 6 is a detailed functional block diagram of a specific embodiment of a central controller 24 (see FIG. 1) according to the invention. The central controller 24 ties together and integrates all lottery system functions providing centralized control and data collection from remote terminals 20 through the terminal controllers 70. A polling procedure is utilized, whereby the telephone number of each terminal controller 70 is called in sequence, followed by transfer of data from the central controller 24 or a message from the central controller 24 requesting data. Requested data is then sent by the terminal controller 70 to the central controller 24 on the telephone line. The central controller 24 also keeps some phone lines available to receive unsolicited exception messages from the terminal controller 24. These messages can call the central controller to replenish exhausted resources in the terminal controller, to report

exceptions, high tier winners and occurrences of the terminal controller operation by the agents. Optionally, encryption processors **181** may be coupled to the computers **180** for encrypting and decrypting transmissions.

The illustrated system is intended to provide high reliability and thus utilizes redundancy and conventional circuitry. It is understood that many other configurations would be possible.

The central controller **24** includes two fully parallel conventional computers **180**, which in the preferred embodiment are Digital Equipment Corporation VAX computers, to perform lottery management, maintenance, and control functions. Two computers **180** are provided for enhanced reliability. However, one computer could alternatively be used. The parallel computers **180** communicate with each other through a high speed interprocessor link **179** which is a conventional computer communications network (e.g. Digital Equipment Corporation's DECNET), enabling them to share resources and constantly monitor each other's functions. At least one conventional hard disk storage device **182** (e.g. Fujitsu "Eagle" M2284 disk drive) is coupled to each of the parallel computers **180**, as shown, for data and file storage. A separate disk storage device **183** is coupled to each computer **180** for logging purposes. In addition, operator control station **184** including a display terminal, teleprinter, and printer, are coupled to the computers **180** to permit operator control and monitoring.

As illustrated in FIG. 6, telephone signals are coupled from the lines **22** to a plurality of modems **178** which are coupled to the computers **180**. Such modems are widely known in the art. In the preferred embodiment, each modem is a standard 300 baud, auto-dial, auto-answer modem.

Although both computers **180** operate simultaneously, when an output is required from the central controller, output from only one is sent to the remote terminal **20**. Thus, only one of the computers is "active" for communication to the terminals but both computers **180** continuously duplicate all functions with only the active output used. The high speed link **179** between the computers **180** provides for the integrity of data going out and permits resolution of control to determine which computer will be active.

In normal operation, each computer **180** sends an "I'm alive-are you" signal to the other on a continuous basis. If the active computer fails, then after a predetermined period during which the inactive computer does not receive an answer, the inactive computer takes over and sends a failure message to the operator console. If the inactive computer fails, the active detects the failure and sends a failure message on the operator console. The central controller **24** then operates in a simplex mode wherein only one computer is operational.

A major function of the central controller **24** is collection of data from the terminal controllers **70**. The central controller **24** maintains a list of phone numbers for each terminal controller **70** and dials these numbers according to a polling algorithm and schedule. The phone numbers, polling algorithm and schedule can be changed from the central controller **24** by the operator.

When either the central or terminal controller has initiated a phone connection with another receiving controller, it sends a connect request message with its password. The receiving controller then sends a connect acknowledge message with its password and waits for a transmission. The initiating controller may then

transmit the appropriate data which must then be acknowledged by the receiving controller. If the receiving controller does not recognize the initiating controller's password, it sends back a connect reject message. The initiating controller may then re-initiate the call. This protocol establishes and authenticates a connection.

The data normally transmitted to the central controller in response to a poll is the game meter data. This includes number of games played, plays cancelled, low tier prizes, current position in mini-pool and free games played by repairmen.

Certain exception messages can also be transmitted to the central controller **24**. These include coin-in jammed, ticket printer paper jammed, ticket printer out of paper, remote terminal housing **32** alarm, premises alarm, and power failure.

The terminal controller **70** can initiate a number of unsolicited messages to the central controller **24**. These messages include cash box door openings, requests for new mini-pools, high tier winner requests and others.

All meter data, exception messages, and unsolicited messages are stored at the central controller **24**. The central controller **24** maintains in its on-line data-base an on-going log of all messages received and maintains an on-line data base so that inquiries may be made regarding specific remote terminals **20**.

The central controller **24** performs a number of major lottery management functions including lottery prize pool maintenance and lottery performance reporting. The central controller **24** controls the prize pools for each game. At the start of each mini-pool of 1000 plays for each game resident in each remote terminal, communications is established with the central controller. The remote terminal **20** first determines its portions of the seed numbers from independent changing events occurring in each remote terminal **20** (e.g. an internal clock value). The remote terminal **20** determined portions of the seeds are then encrypted and transmitted to the central controller **24**. They are decrypted, combined with central controller **24** determined portions of the seed numbers, then re-encrypted, transmitted back to the remote terminal **20**, decrypted, and stored in the sealed terminal controller **70**. The terminal controller processor **150** then controls the distribution of the prizes based on the seeds. New prize pool data is sent to the terminal controller **70** whenever it has exhausted its mini-pool and requests a new mini-pool from the central controller. Thus, the central controller has control of lottery prize pool parameters.

Lottery performance reporting allows for on-line and off-line reports. On-line reports include full remote terminal activity reports, pool status reports, remote terminal availability reports and financial analysis of pool status for the overall lottery system. Off-line reports include, inter alia, prize liability reports, sales summaries, monthly remote terminal activity reports, weekly agent sales reports, and weekly remote terminal availability reports.

The central control of the lottery system permits a number of unique system capabilities. One such capability is an electronic market survey. The sophisticated centrally controlled lottery system can draw a random sampling of players who can be asked to participate in an electronic marketing survey. In the electronic marketing survey a free game play is offered on the remote terminal if the player will answer a few, simple market survey questions. Thus, on a random, or other basis, a

predetermined number of plays in each mini-pool are selected as market survey free plays. The remote terminal displays on the video monitor the offer of a free game in exchange for answers to the market questions and allows the player to accept or reject the offer. Assuming the offer is accepted the basic questions, preferably yes-no questions, are displayed beginning with marital status, and sex, followed by questions about lottery use, level of education, age, location, etc. The player answers the questions using the player control devices 57. A speech recognition unit is particularly suitable for input of survey answers. This market survey could also be sold to others to permit market surveys relating to other than lottery markets. At the conclusion of the questions, a free game is provided.

The programs for the central controller operate within a multi-programming operating system. The operating system utilized in the preferred embodiment is Digital Equipment Corporation's VMS system. There are two categories of processes involving concurrently operating, interactive tasks, communication processes and file service processes. The communications processing includes all activities involving communications to and from the remote terminals. The file services tasks involve file handling interactively coupled to the communications program.

The communications program establishes communications with the terminal controller 70 for the purpose of polling, diagnostics, or dispatching commands. This occurs in response to an appropriate request from a file server program. The communications program also responds to the terminal controller requests for the purpose of providing prize validation, handling exceptions, handling mini-pool requests and handling door opening messages, and other terminal controller information. In addition, the communications program handles encryption and decryption either by utilizing an encryption/decryption subroutine or by utilizing encryption processors attached to the computers 180.

Thus, when a message is received the communication program processes the information, sending it to the appropriate file server program. Conversely, outgoing message data from the central controller 24 is coupled from a file server program to the communications program, which then handles the communication of the data to the appropriate remote terminal 20.

The file server programs are constantly running in the system, serving requests from the communications program and maintaining common files. The file server programs include the following programs:

A polling program controls remote terminal polling, polling to the terminal controller 70 to collect marketing and accounting data. The polls are conducted a predefined number of times a day in scheduled fashion, while twenty percent of them are selected randomly and polled at random times.

A marketing and accounting program handles marketing and accounting data which is sent to the central controller as a result of polls, or which is sent with unsolicited transmissions from the terminal controller 70 (e.g. with mini-pool requests, validations, exceptions, etc.).

A pool program handles creation of new pools and mini-pools on request from the terminal controller.

A validation program processes and stores all the exception messages as they arrive from the terminal controllers.

An operator program allows the central controller operator to access the system handling messages initiated by the operator.

A market survey program accepts and stores the market survey information from the terminal controller.

A hall of fame program receives information about big winners and creates a hall of fame data file. This hall of fame can subsequently by dispatched to the terminal controllers along with the poll requests. The remote terminal can then display the data.

A door checking program processes incoming messages from the terminal controllers carrying information about which remote terminal doors were opened or closed, and in what sequence. This program can create exception messages depending upon the sequence of door opening and closing. In addition, the program also processes information about the contents of the cashbox and handles agent invoice and accounting messages.

A logging program logs into a separate file all incoming and outgoing messages together with all other pertinent information.

Since the central controller 24 is comprised of two identical computers with identical files, each computer runs identical software. All incoming data is routed to both computers, but only one sends out its data. The systems are arbitrated by a watchdog program which runs continuously in each computer and interfaces with the watchdog program in the other computer to determine which computer is active. The results of each computer would also be compared to detect an error condition. The program monitors its own environment by analyzing error messages and by a periodic handshake with the other watchdog program. If it detects that the other system is not in operation, it instructs the communications program to switch to active if it is not already active.

A specific embodiment of the illusion of skill game machine has been described for purposes of illustrating the manner in which the invention may be made and used. It should be understood that implementation of other variations and modifications of the invention in its various aspects will be apparent to those skilled in the art, and that the invention is not limited thereto by the specific embodiment described. It is therefore contemplated to cover by the present invention any and all modifications, variations, or equivalents that fall within the true spirit and scope of the basic underlying principles disclosed and claimed herein.

What is claimed is:

1. In a gaming system the object of which is to win a prize by chance, a video game apparatus which provides an illusion that skill is required to win a prize comprising:

    means for displaying a plurality of symbols on a screen;

    input means operable by a player for initiating a game and for controlling one of the symbols on the screen to interact with at least one of the other displayed symbols;

    means responsive to the initiation of a game for randomly determining whether the player is to win a prize; and

    video game control means responsive to the input means for controlling the non-player controlled symbols to provide a video game presentation wherein the object of the video games is for the player controlled symbol to interact with at least one of the other symbols in a predetermined man-

25

ner, said video game control means being responsive to a win determination by said random determining means for manipulating at least one non-player controlled symbol on the screen to provide an indication of a win on the display.

2. In a gaming system the object of which is to win a prize by chance, a video game apparatus which provides an illusion that skill is required to win a prize comprising:

    means for displaying a plurality of symbols on a screen;

    input means operable by a player for controlling one of the symbols on the screen to interact with at least one of the other displayed symbols;

    video game control means responsive to the input means for providing a video game presentation wherein the object of the video game is for the player controlled symbol in a predetermined manner; and

    means for randomly determining a winner of a prize, said random means being coupled to the display means to cause one or more symbols to be displayed which represent a win wherein the display of said winning symbol or symbols appears to relate to the skill of the player in controlling the player controlled symbol wherein said random determining means is responsive to the video game control means to cause the display of the winning symbol or symbols after the player controlled symbol interacts with another symbol in said predetermined manner so that the display of the winning symbols appear to relate to the skill of the player.

3. The system of claim 2 wherein said video game control means includes means for controlling at least one of the non-player controlled symbols to interact

26

with the player controlled symbol in said predetermined manner.

4. In a gaming system the object of which is to win a prize by chance, a video game apparatus which provides an illusion to a player that skill is required to win a prize comprising:

    means for displaying a plurality of symbols on a screen to provide a video presentation of a game;

    input means operable by a player to control one of the symbols on the screen to interact with at least one of the other displayed symbols;

    control means responsive to the input means for determining the occurrence of a predetermined type of interaction between the player controlled symbol and another symbol, said control means being coupled to the display means to cause a prize value to be displayed on the screen in response to a determination of said occurrence;

    means for randomly determining whether a player is to win the prize value displayed on the screen; and

    means responsive to a win determination by the random determining means for manipulating one or more symbols displayed on the screen to cause the player to win the prize value.

5. The gaming system of claim 4 wherein the magnitude of the prize value is independent of the player's skill in controlling the one symbol to interact with the other symbols in said predetermined manner.

6. The gaming system of claim 4 wherein said manipulating means is responsive to said random determining means to cause the control means to display a matching prize value upon a subsequent occurrence of the predetermined type of interaction, a player winning the prize value in response to obtaining matching prize values.

* * * * *

40

45

50

55

60

65

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,582,324

DATED   : April 15, 1986

INVENTOR(S) : John R. Koza, et al

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 24, line 66, Claim 1, change "games" to --game--.
Column 25, line 18, Claim 2, after "symbol" insert --to
interact with at least one of the other displayed symbols--.

Signed and Sealed this

Eighteenth Day of November, 1986

Attest:

DONALD J. QUIGG

Attesting Officer   *Commissioner of Patents and Trademarks*